NESTE NÚMERO: RESULTADO DO CONCURSO DE FRASES

MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA

Cr$ 12.900

25

Spectravideo 318

As maravilhas do CAD

Computadores ópticos

rioGráfica

MANAUS, ALTAMIRA, MACAPÁ, JI-PARANÁ, PORTO VELHO, RIO BRANCO, SANTARÉM, BOA VISTA, SINOP, ALTA FLORESTA: Cr$ 16.770

MICROCOMPUTADOR CURSO PRÁTICO

RESULTADO DO CONCURSO DE FRASES

No dia 14 de outubro de 1985, na sede da Editora Rio Gráfica Ltda. — Rua Itapiru, 1209, CEP 20251, Rio de Janeiro (RJ) —, realizou-se a apuração e proclamação dos vencedores do concurso de frases promovido pela coleção em fascículos MICROCOMPUTADOR-CURSO PRÁTICO. De acordo com o regulamento do concurso, cada vencedor receberá um microcomputador Sysdata IV. O direito aos prêmios prescreve em 180 dias, contados a partir da apuração. Relacionamos a seguir os nomes dos ganhadores e suas frases.

ROQUE GÖLLER
Rua Andradas, 731, ap. 201
90000 - Porto Alegre (RS)

"A alma do seu **Microcomputador** é você. Estude. **Curso Prático**".

ANTÔNIO FERNANDO ROSA DINI
Rua Luís Beltrão, 159, ap. 105 - Vila Valqueire
21321 - Rio de Janeiro (RJ)

"**M**áximo de **i**nformática **c**om **r**apidez e **o**bjetividade? **Micro**computador - Curso Prático."

MARCOS PERAZZO TAVARES CAVALCANTI
Centro de Ciências Agrárias - UFPB, Campus III
58397 - Areia (PB)

"Microcomputador - Curso Prático e você! Uma dupla programada para vencer."

EMÍLIO HORTMANN NETO
Rua Saldanha da Gama, 25, ap. 401
80000 - Curitiba (PR)

"Microcomputador - Curso Prático... O futuro está à solta nas bancas."

TARCÍSIO RODRIGUES DE ALCÂNTARA
Destacamento da FAB - Aeroporto
57000 - Maceió (AL)

"Microcomputador - Curso Prático: possui a maior memória e a melhor linguagem."

TESTE

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

1) O que significa a sigla CAD?
2) Na tentativa de dotar os robôs de sentidos semelhantes aos do homem, o que substitui os ouvidos?
3) O usuário de micro pode se ver às voltas com cansaço visual ou dor nas costas. Qual a ciência que procura projetar o ambiente de trabalho de forma a evitar tais problemas?
4) Qual a cor de tela considerada ideal pelos usuários de micros?

RESPOSTAS DO TESTE ANTERIOR

1) O fenômeno orgânico que possibilita o funcionamento dos detectores de mentira bem como o "acoplamento mental" do homem ao computador é a RGP (Reação Galvânica da Pele), pela qual as mudanças no estado emocional refletem-se na condutividade elétrica da pele.
2) Os motores que produzem movimentos extremamente precisos, muito utilizados nos robôs, são chamados motores a passo, ou graduais.
3) Os micros padrão MSX adotaram o microprocessador Z80.
4) Os dois primeiros micros padrão MSX introduzidos no Brasil foram o Expert, da Gradiente, e o Hot-bit, da Sharp.

PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 30 fascículos de periodicidade semanal. Os 29 primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 30 incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores.
As duas primeiras capas são descartáveis; as 3.as e 4.as trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 30 (**Volume 2** e **Jogos**).
Volume 1 - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.
Volume 2 - Guardas; frontispício (que virá com o fascículo 30); miolos dos fascículos de 16 a 30; e guardas.
Jogos - Guardas; frontispício e sumário (que constituem as 3.as e 4.as capas do fascículo 30); 3.as e 4.as capas dos fascículos de 1 a 29; e guardas.

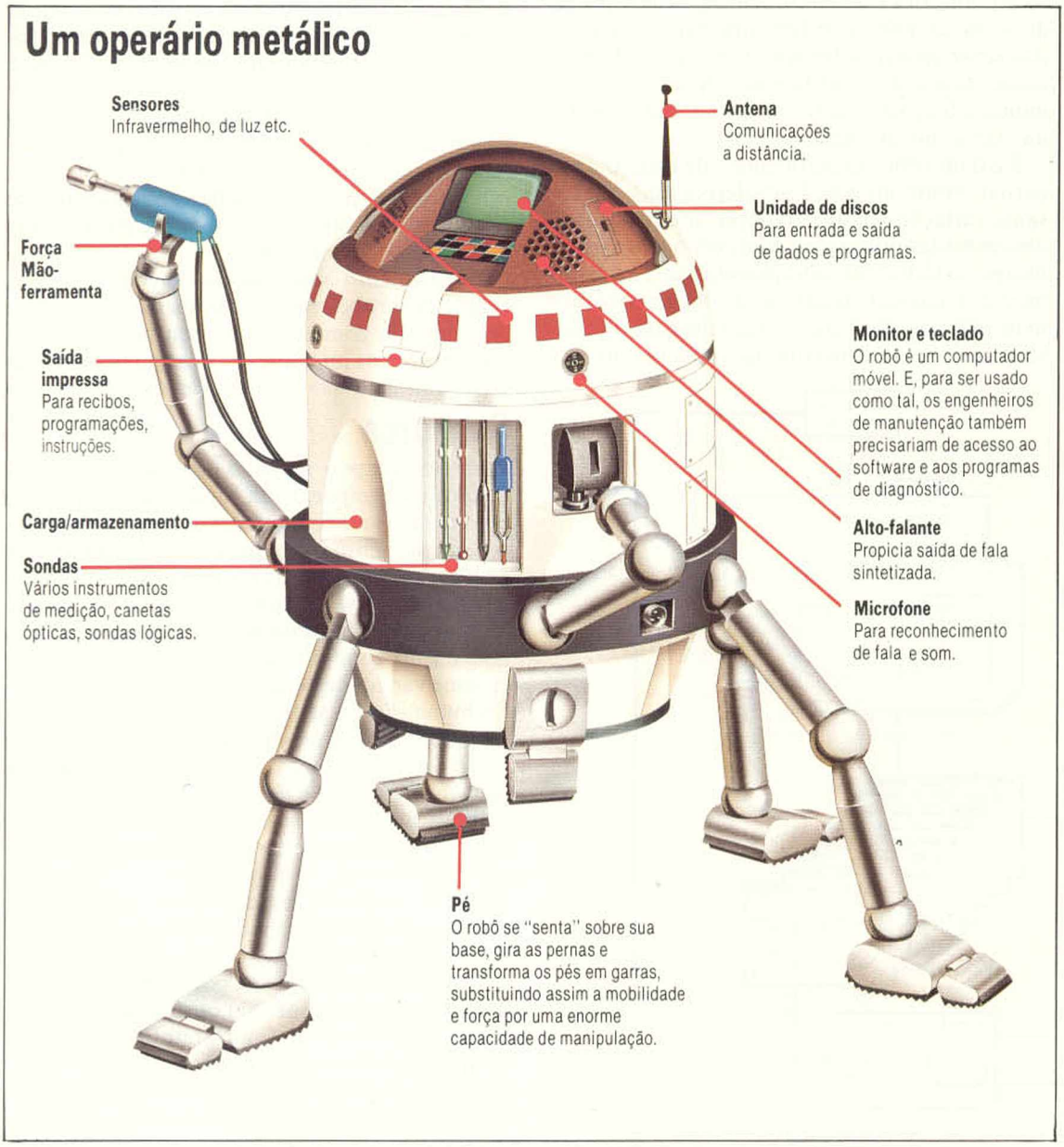
rioGráfica **CONSELHO DE ADMINISTRAÇÃO:** Roberto Irineu Marinho; João Roberto Marinho; José Roberto Marinho; Oscar Neves; **Diretoria:** Oscar Neves; Danilo Esteves Costa; João J. Noro; Marcos March; Nilo Sérgio de Almeida; **GRUPO EDUCAÇÃO E CULTURA - Divisão Comercial - Diretor:** Jaime Rodrigues; **Gerente de Planejamento:** João de Azeredo Coutinho; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgard Mello M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto; **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Maurício Rittner; **Diretor de Arte:** Adalberto Cornavaca; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Editor-assistente:** Claudio Marcondes; **Coordenadora de Revisão:** Isis Augusta Loyolla; **Revisores:** Cacilda Guerra; Luiz Vicente Vieira Filho; Maria da Graça Mendonça Couto; Sérgio Roberto Torres; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistentes de Arte:** Miguel Luis Escámez Simón; Edemilson Ribeiro da Silva; Wagner Celestino; **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; **Texto:** Carlos Eduardo Silveira Matos; José Antônio Arantes; Jussara Maturo; Paulo Brito; Solange A. Menezes; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Luiz Roberto de Godoy Vidal; Maria Clara Cescato; Maria Ester Menezes Martinho; Fernando Dorfman Knijnik; Newton Roberval Eichemberg; **Pesquisa:** Stella Maria Quentel; **Assessoria Técnica:** Geraldo Coen; Gustavo José Ferreira Grubba; José Rubens Salles Toledo; Lauro de Lauro; Pierluigi Piazzi; **Estúdio:** Vagner Facuri de Oliveira. As fotos não creditadas pertencem à obra original. © APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa. **Composição:** Editora Rio Gráfica Ltda. e AM Produções Gráficas Ltda. **Impressão:** JBIG.

NÚMEROS ATRASADOS - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electroliber Ltda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.

PERFIL DO FUTURO

Um operário metálico



Tempo futuro
Tudo indica que ainda teremos robôs não especializados, de aparência humana, a um custo capaz de torná-los os substitutos do trabalho humano semi-especializado. Mas eles precisarão de uma "inteligência" altamente desenvolvida que inclua um banco de dados de conhecimentos e habilidades técnicas, integração sensorial e software de aprendizado.

Entre os vários aspectos relativos aos modernos robôs estão suas limitações práticas. Mas as conquistas tecnológicas já apontam os possíveis desenvolvimentos futuros.

Os robôs reais estão muito longe daquelas máquinas pensantes concebidas pela ficção. Nossa imaginação nos condicionou a esperar deles determinadas qualidades: que sejam capazes de se mover à vontade; de ver, ouvir e sentir as coisas ao seu redor; de conversar conosco sobre filosofia e ciência ou, pelo menos, de comunicar-se de modo inteligente; e de manipular objetos e idéias tal como fazemos. Em outras palavras: nós os criamos à nossa imagem e semelhança. Assim, quando examinamos criticamente os robôs usados para fins comerciais, industriais ou amadorísticos, muitas vezes nos admiramos de como desempenham suas tarefas específicas, ao mesmo tempo que nos decepcionamos por não conseguirem fazer nada além disso. Partindo do que conhecemos hoje sobre projetos e implementação de robôs, o que podemos esperar — em termos realistas — dos robôs do futuro?

Variações visuais
O conhecimento de um objeto pode ser armazenado como um "gabarito" do original, juntamente com dados sobre sua variação; cada variação pode ser aplicada ao gabarito para produzir uma imagem diferente, mas que ainda se encaixa numa definição do original. Um módulo de análise geral esquadrinha a imagem proveniente dos sensores do robô, possibilitando uma indicação sobre o tipo genérico do objeto. Todas as variações são então comparadas com a imagem recebida até que haja uma correspondência. A certeza estatística com que se obtém essa correspondência determina se as novas variações da imagem em relação ao gabarito apresentam-se tão profundas a ponto de exigirem uma nova instrução de variação de dados.

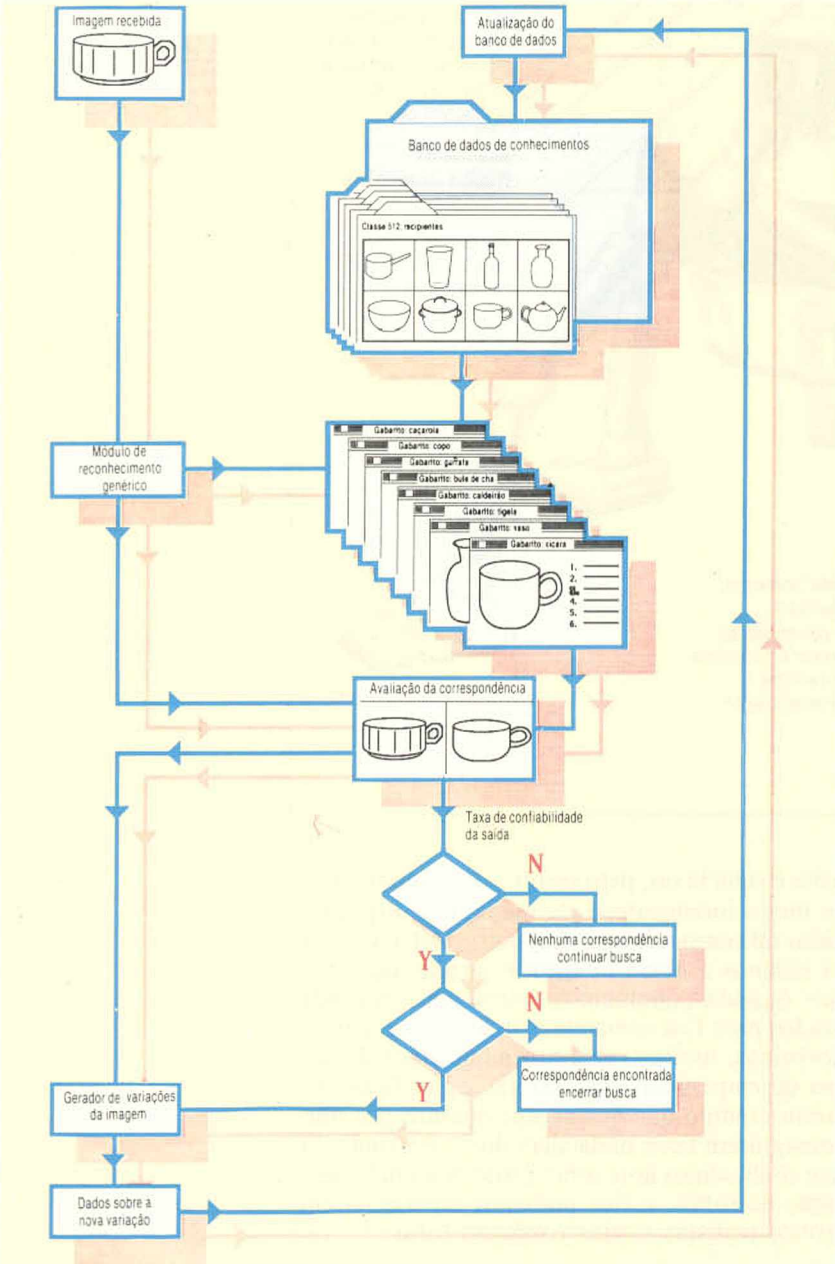
Movimento

É pouco provável que, num futuro próximo, os robôs venham a se locomover sobre algo parecido com as pernas humanas. Seria necessário dedicar muito tempo e espaço de processamento ao esforço de manutenção do equilíbrio, já que as articulações e a "musculatura" elétrica e hidráulica do robô não têm flexibilidade igual à dos seres humanos (graças à interação de músculos, tendões e cartilagens). Além disso, em muitas situações, o deslocamento sobre duas pernas seria um obstáculo.

Existem robôs experimentais de quatro e seis pernas, como insetos. Isso oferece uma interessante variação ao projeto dessas máquinas para determinadas utilizações. Mas, em operações militares, explorações interplanetárias e usos domésticos convencionais, as rodas se mostram mais práticas. No futuro, o movimento do robô será mais flexível, mas jamais poderá competir com a desenvoltura e naturalidade de um atleta em ação.

Robôs industriais e braços mecânicos são, na maioria, necessariamente fixos, com movimentos limitados a uma área de ação bem específica, já que foram projetados para desempenhar apenas uma ou duas tarefas. Mesmo que se modifique de maneira radical a estrutura das linhas de montagem, é provável que, ainda assim, robôs industriais com capacidade de se deslocar permaneçam relativamente confinados: seguirão linhas demarcadas, correrão sobre trilhos, deslizarão sob suportes aéreos. Pode ser que os aperfeiçoamentos na automação e no projeto de robôs tragam mudanças radicais nos métodos de produção industrial, mas não dá para prever quais serão tais mudanças.

Um elemento-chave do movimento é a necessidade de o robô reagir ao ambiente. Isso implica equipá-lo com elementos sensores, devendo essa entrada de informações estar diretamente relacionada com sua capacidade de movimento.

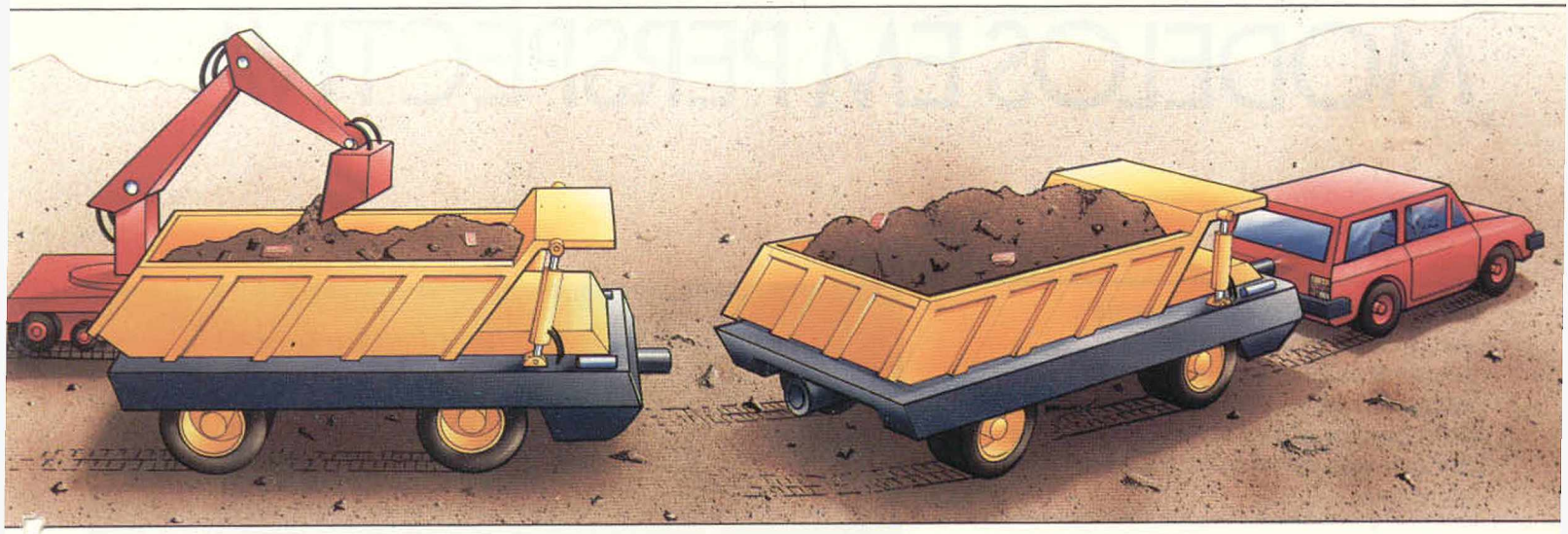


Sensores

Podemos adaptar ao robô equipamentos sofisticados que ampliem sua capacidade de percepção em áreas novas e mesmo desconhecidas para o homem. Sensores de proximidade, detectores de movimento, dispositivos precisos de realimentação posicional descontínua, captadores de ruídos numa ampla gama de freqüências etc., tudo isso dá ao robô capacidade para coligir uma enorme variedade de dados. Bem mais que os seres humanos. Os sistemas visuais vêm adquirindo precisão cada vez maior, com melhoria crescente na resolução das imagens percebidas. As técnicas de síntese e de reconhecimento da voz, embora ainda em desenvolvimento, tendem a maior sofisticação e terão lugar importante no aperfeiçoamento da robótica.

Robôs utilitários, especialmente na indústria, continuarão a ser equipados apenas com os sensores necessários para execução de tarefas específicas. Braços soldadores, por exemplo, prescindem da capacidade de fala ou de realimentação visual. Eles executam suas tarefas com precisão e rapidez, tendo um mínimo de entrada sensorial, e as percepções irrelevantes talvez só viessem perturbar-lhes o desempenho.

Os robôs não especializados, projetados para simular processos do pensamento humano e aprender com a experiência, teriam de ser equipados com o maior número possível de sensores. O fundamental seria que pudessem investigar o ambiente com autonomia e assimilar as informações que coligissem. Os seres humanos apóiam-se numa combinação de informações visuais e auditivas para compreender a fala. Muitas vezes tendemos a ignorar essa síntese, principalmente quando pensamos em termos de projeto de robôs. Mas, para que eles possam se comunicar com o homem, compreender sua linguagem, em vez de apenas reconhecê-la, seria essencial conjugar visão e audição.



Hoje, o volume de dados que os robôs recebem e com o qual podem lidar é restringido pela quantidade de memória necessária para armazenar as entradas sensoriais e pelas limitações na capacidade e velocidade de processamento. Um robô é capaz de armazenar a imagem visual de um objeto — uma maçã, por exemplo — e vincular essa imagem a um nome. O armazenamento da imagem ocupa espaço na memória e, quanto maior a resolução visual, maior a quantidade de memória requerida para manter a imagem. Como as maçãs não são todas iguais, o robô deverá ter capacidade suficiente de memória para armazenar vários modelos representativos de imagens de maçãs. Ou, então, um algoritmo que identifique as variações e gire a imagem básica, de modo que a maçã seja reconhecida em qualquer posição. Mesmo com um nível mínimo de resolução (digamos, 256 pixels por imagem), o número de variações pode chegar à casa dos milhares.

Futuramente, as exigências de memória talvez venham a ser preenchidas pelos chips RAM de grande capacidade (chips de 1 Mbit já se encontram em desenvolvimento) e pelos chips RAM interativos, que armazenam dados "variacionais". Os dados não específicos mantidos nesses chips, sempre que necessário, podem ser chamados pelo processador, para elucidar imagens diferentes, quase como uma sub-rotina num programa em BASIC.

Afora isso, a verdadeira percepção sensorial do robô exigiria que os dados chegassem de várias fontes ao mesmo tempo e que fossem rapidamente executados. Os processadores existentes seriam incapazes de lidar com um volume tão grande de informações simultâneas. Assim, os dados logo começariam a se acumular, entrando em ordem de espera. Uma provável solução estaria no uso de dois ou mais processadores de alta velocidade e capacidade operando paralelamente. Nesse caso, um processador de controle funcionaria como gerenciador, distribuindo tarefas aos processadores inativos em algum ponto do sistema.

O problema, porém, não está apenas no hardware. Para compreender o objetivo do processamento, o robô precisará de um software muito complexo. Isto é, uma verdadeira "mente".

Mente

Como já vimos, duas são as direções principais da robótica. A que tem maiores possibilidades é a área de instrumentos inteligentes, ou robôs utilitários. Braços mecânicos e sistemas de fabricação automatizados que executam tarefas específicas só requerem um equipamento de software de controle cuidadosamente definido. O braço receberia então um conjunto de coordenadas e uma programação para executar uma seqüência de ações, sem precisar saber o que faz, onde está ou qualquer detalhe sobre o ambiente.

À medida que os robôs são solicitados a executar uma variedade mais ampla de tarefas, porém, estas têm de ser, necessariamente, menos definidas. Se eles tiverem de se movimentar numa sala onde a posição dos móveis se modifica todos os dias, precisarão não apenas coligir e processar informações, mas também incorporar novos elementos em sua percepção do mundo. Seria imprescindível um software de controle operacional expansível segundo a solicitação.

A criação de uma mente mecânica suscita importantes questões, até agora sem respostas, a respeito de como o homem pensa e aprende. Por exemplo: o que de fato sabe uma criança ao nascer? O adulto é produto do meio ou de sua hereditariedade? E qual a relação entre esses parâmetros? O homem parte de um conjunto de estruturas internas que o ajudam a aprender a linguagem, a matemática e a adquirir sensibilidade estética? Se é assim, como funcionam tais estruturas? Para responder a essas perguntas, precisamos estudar o cérebro humano. No futuro, talvez os robôs nos ajudem no desenvolvimento de uma compreensão maior sobre nós mesmos. Embora essas questões não sejam totalmente teóricas — como experiências nesse sentido vêm atestando —, a idéia de um robô pensante ainda está bastante longe.

Comboio!
As técnicas da robótica exercerão maior impacto sobre a sociedade quando forem incorporadas às ferramentas especializadas de baixo nível tais como guindastes, caminhões basculantes, veículos para entregas locais e transporte pesado.
Na figura, um robô-basculante carrega com terra uma série de robôs-caminhões. Depois de carregado, cada robô-caminhão se move de modo semi-inteligente até um ponto de encontro e se engata a uma fila de outros. O comboio se move então pela força de cada robô-caminhão, mas controlado por um homem dirigindo o primeiro veículo. A união da capacidade humana de decisão e comando com a força e a inteligência de finalidade única dos robôs é a maneira mais barata e lucrativa de unir os recursos de hoje com a tecnologia de amanhã.

MODELOS EM PERSPECTIVA

O emprego do computador na geração de gráficos é um dos segmentos de tecnologia mais avançada na informática. Vamos ver os recursos oferecidos aos usuários por um dos mais sofisticados sistemas: o CAD.

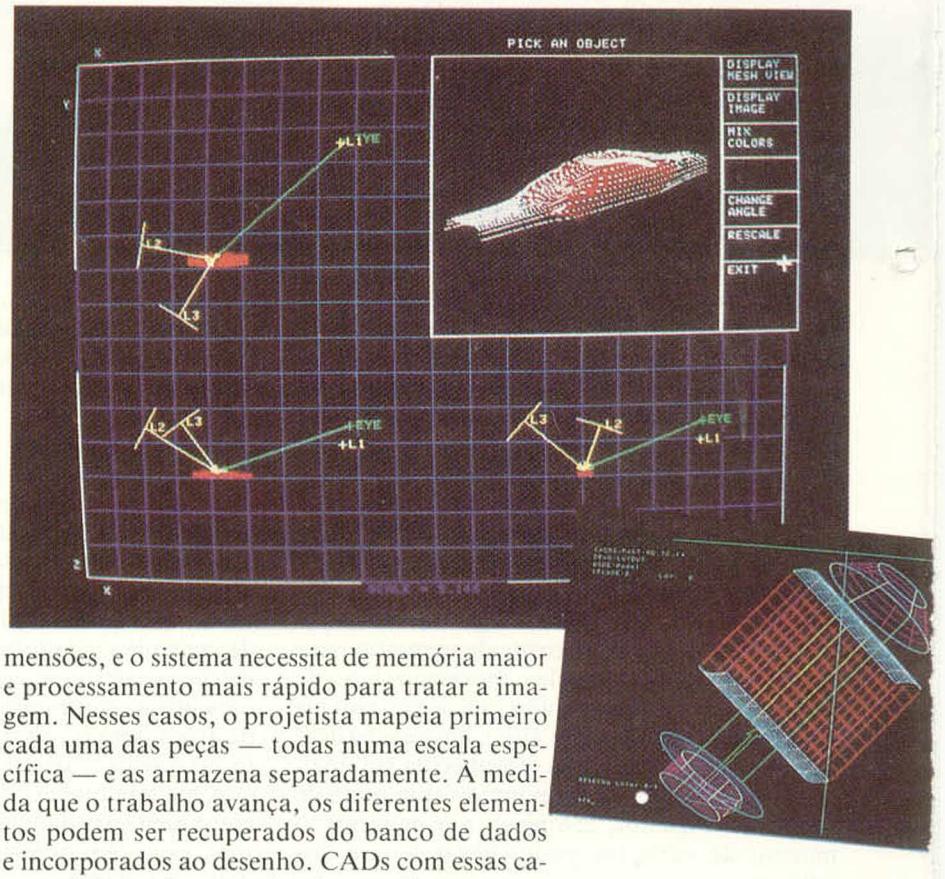
A diferença entre os desenhos elementares em computador e os obtidos com o sistema CAD se evidencia no significado da própria sigla: Computer Aided Design ("projeto com auxílio do computador"). Em sua memória se aloja uma perspectiva real do objeto, e não um simples desenho. Os sistemas CAD mais sofisticados podem projetar um automóvel incorporando cada uma de suas peças e, em seguida, simular tensões e deformações, assegurando-se assim de que ele não se fará em pedaços quando em uso.

Na maioria dos CADs, os desenhos são elaborados de modo diferente dos sistemas gráficos comuns, construindo-se as imagens por meio de pixels (células pictóricas); o CAD utiliza gráficos vetoriais bem mais precisos. Nos gráficos com pixels, uma linha consiste numa série de pontos na tela; já no sistema vetorial, é entendida como a menor distância entre as coordenadas das duas extremidades. O desenho baseia-se na descrição matemática das formas, e a maior precisão permite ao processador trabalhar com resolução muito maior que a do monitor. Tudo isso assegurou ao CAD, sistema criado em 1963, uma utilização crescente nas indústrias mecânica e eletrônica, na engenharia e na arquitetura, na topografia e nas publicações técnicas.

Os CADs mais simples criam desenhos bidimensionais. Algumas vezes podem sugerir uma terceira dimensão — que normalmente se chama de desenho em 2 1/2 D. A "meia" dimensão extra atende às necessidades de um escritório de arquitetura, por exemplo, permitindo a visualização das elevações, planos de base e a análise da declividade de um terreno.

Algumas companhias que projetam circuitos eletrônicos usam sistemas bidimensionais de maior complexidade. Em geral, na eletrônica, o CAD encontra-se associado a programas de verificação. Depois que um projetista de chips mapeia os milhões de circuitos de um microprocessador, o CAD faz a checagem do funcionamento desses circuitos.

O uso mais complexo do CAD prende-se a projetos industriais, que chegam a integrar milhares de peças. Os desenhos requerem três dimensões, e o sistema necessita de memória maior e processamento mais rápido para tratar a imagem. Nesses casos, o projetista mapeia primeiro cada uma das peças — todas numa escala específica — e as armazena separadamente. À medida que o trabalho avança, os diferentes elementos podem ser recuperados do banco de dados e incorporados ao desenho. CADs com essas características são essenciais às grandes empresas automobilísticas. Por exemplo, a Volkswagen do Brasil utilizou as estações gráficas desenvolvidas pela Intergraph americana — fornecedora de tecnologia à Sisgraph brasileira — durante o projeto do automóvel Santana.



Desenhos em luz e sombra

Num sistema CAD, a mais simples forma de desenho tridimensional, a chamada Wireframe, mostra somente os contornos do objeto, sem imagens das superfícies. Sistemas mais sofisticados podem produzir imagens tridimensionais coloridas, com até 256 cores diferentes na tela ao mesmo tempo. Nesses sistemas, é possível produzir imagens de superfícies multicoloridas e sombreadas, opacas ou translúcidas, mostrando os efeitos de múltiplas fontes de luz.

Os usuários de CADs têm a seu dispor uma ampla gama de recursos. Destacamos a seguir os mais importantes.

Dimensionamento. Terminado o desenho, o sistema pode acrescentar automaticamente todas as medidas numéricas, deixando o projeto pronto para apresentação.

Hachuras. Processo de sombreamento automático, usado em desenhos técnicos quando se pretende destacar determinadas partes do projeto.

Superposição. Os projetos podem ser elaborados em camadas. Por exemplo, um arquiteto colocaria a planta de uma casa numa camada e a das canalizações de água e esgotos em outra. As plantas apareceriam em cópias separadas ou na mesma cópia.

Segmentos. Os elementos que compõem o desenho são armazenados num "catálogo" de desenhos prontos, que o projetista pode chamar a qualquer momento.

Expansão/condensação. Esse recurso dá ao projetista a possibilidade de alongar ou encurtar linhas — como se fossem um elástico — para colocá-las onde for necessário.

Ajuste de escala. Permite ao desenhista projetar no tamanho que queira e em seguida armazenar as informações na escala do restante do desenho.

Zoom e pan. Termos tomados por empréstimo ao cinema. No zoom, ampliamos uma parte do desenho até que encha a tela; com o pan, ao contrário, a afastamos, a fim de visualizarmos o conjunto do desenho.

Dispositivos de entrada

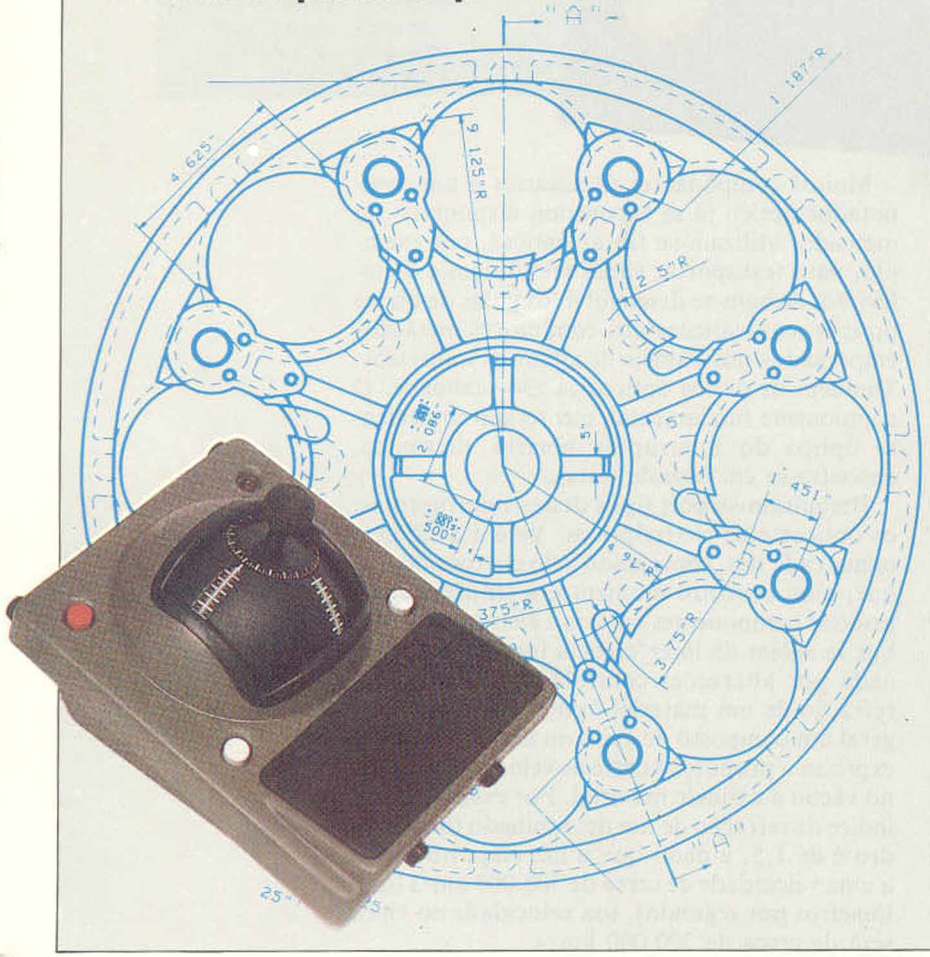
Os usuários de CADs geralmente escolhem os dispositivos de entrada conforme o trabalho a realizar. Entre os mais familiares estão o tablete digitalizador, a caneta óptica, o rastreador (para leitura automática de textos impressos em papel) e o mouse.

Usam-se os processadores de imagens para a introdução de fotografias e desenhos complexos. Eles fotografam a figura a ser introduzida e decompõem a foto em elementos reticulados, que formam uma imagem na tela.

O teclado é um meio eficaz de introduzir as coordenadas dos gráficos vetoriais. Juntamente com o mouse, é um dos componentes do Interpro 32, a estação gráfica produzida pela Sisgraph, com microprocessador de 32 bits, que pode ser utilizada em separado ou ligada a redes de computadores já instaladas.

Também se usam joysticks, só que bem mais sofisticados que os dos jogos eletrônicos. O Bitstik, da Robocom, tem potenciômetros que medem o movimento do controlador com precisão de até seis casas decimais. O joystick comanda um cursor na tela e, com os botões, selecionam-se funções no menu. A extremidade do bastão pode ser ajustada para ampliar um detalhe ou introduzi-lo no desenho completo.

Desenho de precisão por meio do Bitstik



O CAD no micro doméstico

O sistema gráfico CAD torna-se uma ferramenta cada vez mais confiável, para criação e testes de projetos sem a necessidade de se construírem modelos, sempre caros. Muitos sistemas hoje disponíveis destinam-se a micros domésticos de pequena capacidade, embora exijam periféricos e equipamentos adicionais para resultados a nível profissional.

Para funcionar com eficácia, o CAD exige muito software adicional. O enorme volume de informações a processar requer uma grande capacidade de memória. O usuário precisa de um controlador de largura dupla e uma placa de memória, e placas adicionais de memória caso se acrescentem cores. Para aceitar informações do computador MTX, é necessário um conversor A/D "flash" simples (monocromático) ou de três canais (em cores). Um sistema totalmente equipado armazena dados, vindos da câmara de vídeo, à velocidade de 7,2 Mbytes por segundo — e tal potência custa caro.

Já o sistema Bitstik é bem mais acessível. O pacote foi desenvolvido para o BBC Micro e permite a produção de desenhos técnicos de qualidade profissional. Consiste num controlador — uma espécie de joystick de alta precisão —, disco de sistemas, ROM de aplicativos com 8 Kbytes e disco-biblioteca contendo várias famílias de letras e símbolos comuns. Entretanto, para rodar o pacote, deve-se associar o BBC Micro a um segundo processador 6502 e a duas unidades de disquetes de oitenta trilhas. E, para obter o máximo do sistema, o usuário precisará de um monitor colorido e de um traçador, também em cores.

Mas o investimento parece valer a pena: segundo os fabricantes, com esses equipamentos, mesmo pessoas sem inclinação para o desenho produzem gráficos de padrão profissional. A base do sistema é a biblioteca de formatos mantida em disco e as "primitivas" — linhas e arcos armazenados em ROM. A partir das primitivas, elaboram-se conjuntos de diagramas cada vez mais complexos, que então podem ser incorporados a desenhos maiores. A parte principal do sistema está no controlador Bitstik, que permite o posicionamento das primitivas com absoluta precisão.

A LUZ FANTÁSTICA

Recentes avanços no campo da computação óptica possibilitam o uso de feixes luminosos para transmissão de informações a uma velocidade milhares de vezes superior à dos computadores eletrônicos.

Condutor de fibras
O equivalente óptico do fio elétrico é o cabo de fibras. Permite "encurvar" os sinais luminosos e, assim, direcioná-los à vontade. As fibras que preenchem o cabo contêm uma substância transparente que "conduz" a luz na direção da fibra.

Quando pensamos em "computador", em geral visualizamos um aparelho que se liga na tomada, baseado em interruptores binários eletrônicos que armazenam e processam informações. Entretanto, é errado concluir que esse seja o único tipo possível de computador. Teoricamente, um computador pode basear-se em qualquer método de armazenamento de informações: carga elétrica, engrenagens, orifícios num papel, contas, pedras, ou qualquer outro meio.

O semicondutor eletrônico constitui o melhor tipo de armazenamento inventado até hoje, pois consome pouquíssima energia, tem tamanho microscópico e, sobretudo, é rápido. A velocidade de operação do computador fica limitada apenas pelo tempo que seus interruptores binários necessitam para passar de uma posição a outra, e pelo tempo de transmissão de informações entre eles. Circuitos baseados em fluxos de elétrons mostram-se muito rápidos na execução dessas funções. Mas já são possíveis, teórica e tecnicamente, circuitos ainda mais rápidos, baseados em feixes de luz. Existem pesquisas em computação óptica e em disciplinas a ela vinculadas, como a "lógica fotônica".

Muitos componentes necessários a um computador óptico já se encontram disponíveis no mercado: utilizam-se fibras ópticas, por exemplo, para transportar sinais telefônicos e de televisão. Acham-se desenvolvidos guias de ondas ópticos, equivalentes aos condutores metálicos empregados nos painéis de circuitos impressos. Também os discos ópticos já são realidade. O componente fundamental, que seria o equivalente óptico do interruptor binário eletrônico, encontra-se em fase de testes.

Pesquisam-se dois tipos desses interruptores: os ópticos e os eletroópticos. Vale a pena examinar cada um, começando pelo eletroóptico, já que, num primeiro momento, a tendência será associar componentes ópticos e eletrônicos. Ambos se valem da interferência luminosa ocasionada por alterações controladas no índice de refração de um material óptico não linear, em geral um composto de gálio ou lítio. Esse índice expressa a proporção entre as velocidades da luz no vácuo e naquele material. Por exemplo, se o índice de refração de um determinado tipo de vidro é de 1,5, e dado que a luz viaja no vácuo a uma velocidade de cerca de 300.000 km/s (quilômetros por segundo), sua velocidade no vidro será de cerca de 200.000 km/s.

Programa luminoso
Outro componente essencial para a concretização do computador óptico é este dispositivo semicondutor, que emite uma faixa de freqüências de luz pura. Isso será fundamental no tocante às interferências nas comunicações, onde se requer uma padronização na freqüência e no comprimento das ondas luminosas.

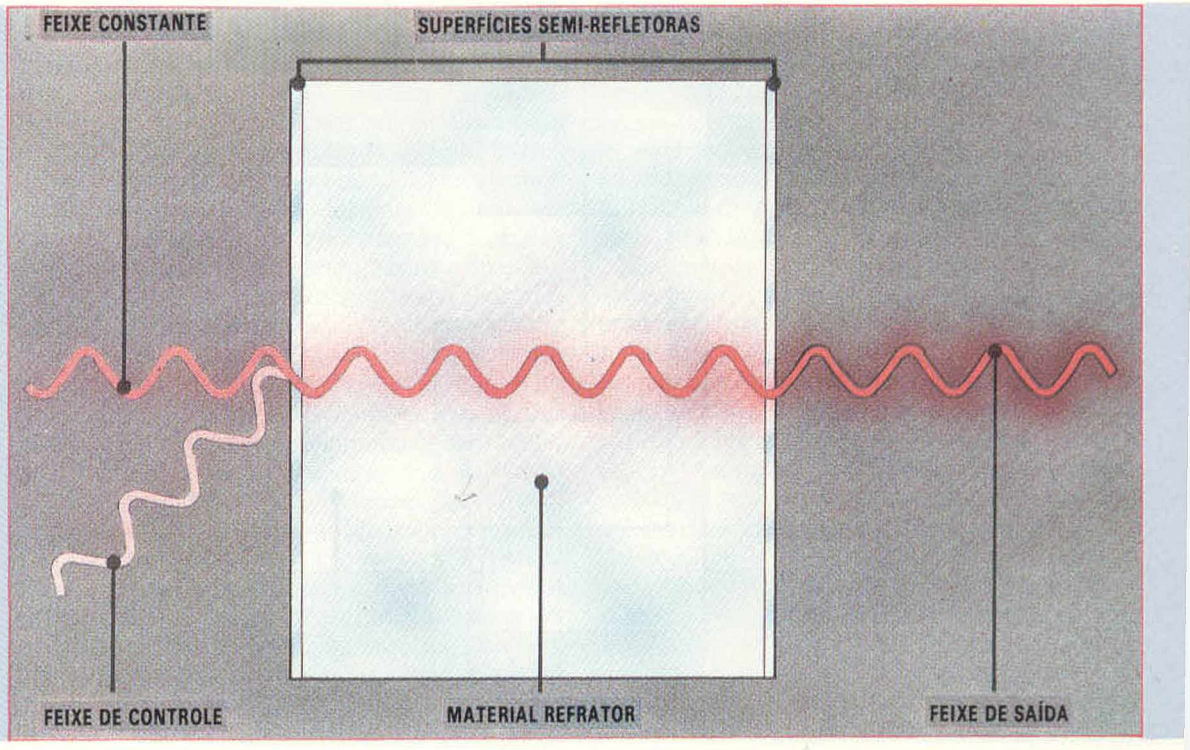
O interruptor binário eletroóptico baseia-se no interferômetro Mach Zehnder, dispositivo cujo princípio de funcionamento é muito simples. Um feixe luminoso que o atinge divide-se em dois, passa por canais independentes, feitos de material não linear, sendo então recombinado. Caso não seja aplicada uma corrente elétrica a um dos canais do interferômetro, os dois raios chegam à saída em fase e geram um forte feixe — o estado "ligado". Se alterarmos, por meio de um sinal elétrico, o índice de refração de um dos canais, os dois atingirão a saída defasados, não se recombinando, pois suas velocidades serão diferentes. Resultará um feixe de saída muito mais fraco — o estado "desligado".

Esse tipo de interruptor constituiria a base de um circuito integrado óptico. Seria mais rápido que um equivalente eletrônico, pois a luz transporta informações a uma velocidade muito maior que a dos sinais eletrônicos (cerca de 10.000 vezes mais rápida, devido à freqüência mais alta das ondas luminosas). Apesar desse desempenho superior, a velocidade do circuito misto ainda fica limitada pela realimentação elétrica, necessária para operar os interruptores. Podem-se obter velocidades ainda maiores usando a luz (um feixe fraco controla outro mais forte).

Esse tipo de estrutura é análoga à do transistor eletrônico. Um protótipo de interruptor inteiramente óptico, chamado transfasor, já foi construído por uma equipe da Universidade de Heriot Watt, em Edimburgo, na Escócia. Esse dispositivo liga e desliga em cerca de 1 picossegundo (1 ps = 1 trilionésimo de segundo, ou seja, 1 ps = 10^{-12} s). Em comparação, o mais rápido transistor eletrônico possui uma velocidade de ligação de 1 nanossegundo (1 ns = 1 bilionésimo de segundo, ou seja, 1 ns = 10^{-9} s). Isso significa que um computador inteiramente óptico operaria a velocidades mil vezes superiores a um eletrônico.

O transfasor desenvolvido na Universidade de Heriot Watt baseia-se num interferômetro inventado pelos físicos franceses Charles Fabry e Alfred Perot em 1896. Nele, coloca-se um material não linear entre duas superfícies refletoras. A luz que atravessa uma delas ricocheteia na camada intermediária antes de sair pela superfície oposta. As ondas de luz interceptadas interagem de uma forma determinada pelo índice de refração

Sobre o feixe luminoso
O elemento básico do computador eletrônico é o transistor; seu equivalente óptico chama-se transfasor. Um feixe constante entra por uma das extremidades, juntamente com um feixe de controle; ambos interferem em conjunto no interior do dispositivo. Uma pequena variação na intensidade do feixe do controle produzirá notável mudança na saída luminosa. Desse modo, o transfasor pode ser usado como interruptor lógico.



do material não linear, que pode ser ajustado por um feixe de controle. Se o índice de refração for tal que as ondas luminosas fiquem em fase e se reforcem mutuamente, o resultado será um feixe luminoso forte; se ficarem fora de fase, o feixe transmitido resultará fraco. Os dois estados de transmissão alternativos correspondem às posições "ligado" e "desligado".

Vantagens teóricas

O sucesso do computador óptico será decidido, em última análise, pelos limites teóricos e técnicos das máquinas convencionais. Em tese ele é superior a seus equivalentes eletrônicos em diversos aspectos, começando pela velocidade de comutação. Para que um transistor funcione como interruptor, uma corrente de elétrons deve atravessar sua base. Há limites para a velocidade de movimento dos elétrons num semicondutor; bem como para a espessura mínima de um transistor (quanto mais fina sua base, mais curto o trajeto dos elétrons). Um computador óptico, transmitindo à velocidade da luz, é inerentemente muito mais rápido.

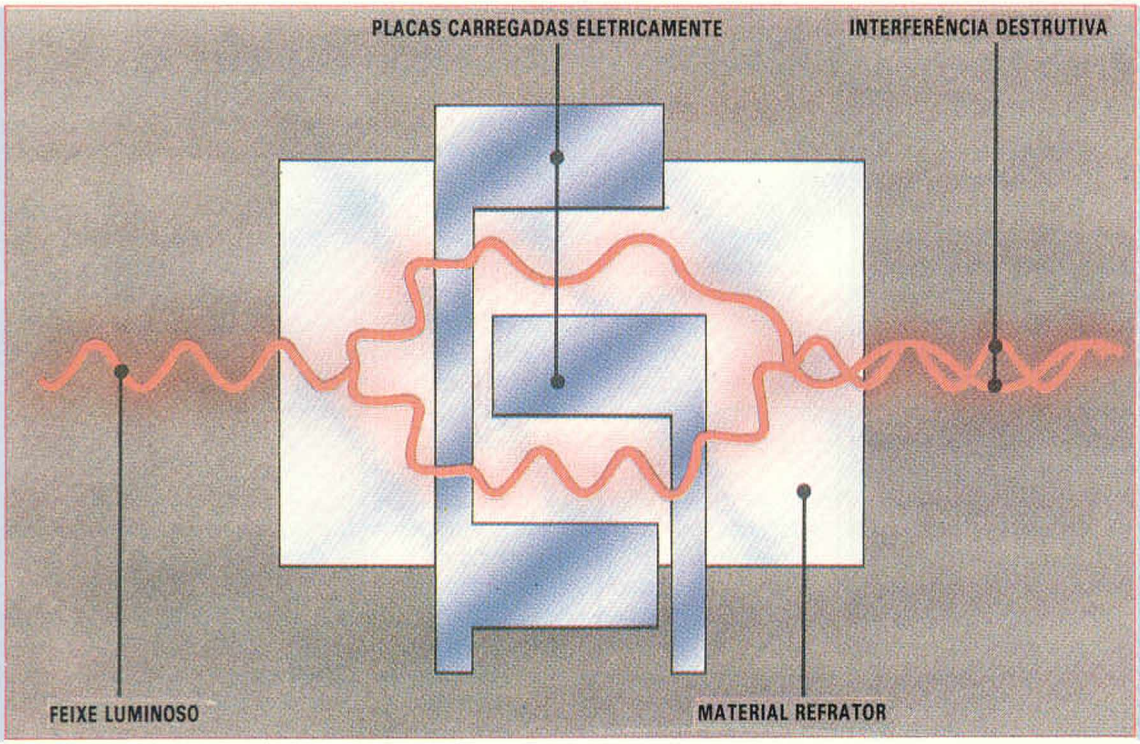
As máquinas convencionais também são limitadas pelo chamado "engarrafamento de Von Neumann". Característico da arquitetura básica de quase todos os computadores, diz respeito ao modo serial de acesso às informações na memória (uma palavra por vez). Um computador óptico acessa a memória de modo paralelo, pois os feixes luminosos não precisam ser isolados, como ocorre com os sinais elétricos. Diversos feixes separados, assim, correm pelo mesmo transistor óptico e operam simultaneamente.

A terceira limitação do computador convencional consiste no pequeno espaço disponível para conectar fios num chip. Poucas conexões são possíveis, devido ao problema do leque de pinos — limite quanto ao número de pinos que podem ser montados numa fina camada de silício. O uso de técnicas tridimensionais holográficas (ópticas) possibilitaria mais liberdade no sistema de fiação e permitiria modificar instantaneamente a configuração das conexões.

Finalmente, há vantagens práticas na interconexão dos computadores ópticos e dos periféricos. Já se usam redes de fibra óptica para ligar dispositivos eletrônicos, sendo as mensagens luminosas extremamente rápidas, além de se poderem transmitir várias delas na mesma fibra sem interferência mútua. Essas redes conectam vários computadores ópticos sem necessidade das lentas interfaces optoelétricas.

Em vista dessas vantagens, para quando podemos esperar o surgimento de computadores ópticos práticos e financeiramente acessíveis? Eis a resposta: eles ainda demorarão algum tempo. As vantagens da óptica sobre a eletrônica são maiores na teoria do que na prática. Os crescentes avanços na miniaturização dos chips significam que, pelo menos a curto prazo, os computadores ópticos estão perseguindo um objetivo cada vez mais alto. O número de circuitos num único chip vem dobrando a cada dezoito meses. Chips com 10 ou 100 milhões de circuitos estão previstos para o ano 2000. A chamada LSI (Large Scale Integration, "integração em larga escala"), transformou-se em integração a altíssima (e ultra) escala. O "processador paralelo" é hoje "maciçamente paralelo".

A tecnologia eletrônica de alta velocidade encontra-se em contínuo desenvolvimento, sob dura competição. Visto por esse prisma, o computador óptico a custo acessível pode estar a anos-luz de distância.



Fase a fase
O interferômetro Mach Zehnder permite o funcionamento conjunto de dispositivos eletrônicos e ópticos. A luz que entra por uma das extremidades divide-se em dois feixes. Após a entrada dos feixes no dispositivo, ajusta-se o índice de refração em torno de um deles. A velocidade da luz diminui, ficando então defasada em relação ao outro feixe. Quando os dois feixes convergem, na saída, as fases terão uma interferência destrutiva, resulta um "desligado", que pode ser considerado como um zero lógico.

SPECTRAVIDEO 318

O micro Spectravideo incorpora muitos elementos do padrão MSX, planejado por japoneses e americanos. O ponto alto desse equipamento, além de seu baixo custo, está nos recursos de expansão que oferece.

O Spectravideo 318 é um microcomputador de baixo custo que possui a mesma qualidade do BBC Micro e do Commodore 64. Incorpora sintetizador de som, gráficos de alta resolução com sprites, joystick embutido e entrada para cartucho. O teclado tem muito em comum com o do MSX e design semelhante ao do Sinclair Spectrum. Como este, suas teclas também possuem toque suavizado, e a distância entre elas torna sua operação muito mais agradável.

As teclas de comando do cursor transformaram-se num joystick embutido. Um disco substitui as costumeiras quatro flechas: tocando o disco na área apropriada, movimenta-se o cursor para cima, para baixo, para a esquerda ou para a direita. Mas pode-se encaixar, em lugar do disco, um joystick, que, ao ser acionado, produz o mesmo efeito das teclas; também é empregado na correção de um programa, posicionando o cursor sobre o erro a corrigir.

O BASIC do 318 é o mais atual de uma linha de interpretadores Microsoft. Essa versão aproxima-se do BASIC MSX, uma extensão do BASIC GW, usado em computadores como o IBM PC. É fácil elaborar programas nesse BASIC, que conta ainda com recursos genéricos, como a renumeração das linhas ou sua numeração automática. Embora não possua os novos comandos "estruturados", como WHILE-WEND e REPEAT-UNTIL, presentes em muitas das novas versões do BASIC, possui o IF-THEN-ELSE, que ajuda a escrever programas claros e eficientes. Esse BASIC tira o máximo proveito da capacidade gráfica do 318.

A tela possui razoável resolução, 256 x 192 pixels e dezesseis cores, embora pequenos grupos de pixels compartilhem as mesmas cores. Esses recursos podem parecer pobres se comparados aos de outros micros, mas é o nível de controle sobre o vídeo que possibilita bons gráficos e não

Teclado do Spectravideo
O teclado do 318 se aproxima muito das especificações MSX. O estilo assemelha-se ao do Spectrum, mas o espaçamento e a qualidade das teclas tornam perfeita sua utilização. É um teclado muito bem equipado, com todas as teclas necessárias: [Tab], [Control], [Escape] e [Backspace], mais cinco para funções (cada uma duplicada), uma [Stop], outra [Select], e um conjunto para edição [Ins], [Del], e [Copy]. Um conjunto de formas gráficas (blocos, pontos e linhas) fica disponível ao se pressionar uma tecla alfabética junto com a [Left GRPH] ou com a [Right GRPH]. As formas que cada tecla produz são nitidamente delineadas no teclado.

apenas as especificações de tela. O 318 inclui todas as extensões gráficas do BASIC da Microsoft, como comandos para traçar pontos, retas, retângulos, círculos, arcos e elipses. Um comando PAINT preenche com uma cor qualquer forma geométrica fechada. As instruções especiais VPOKE e VPEEK permitem um controle ainda maior para ler e gravar diretamente na memória de tela. Há também uma minilinguagem gráfica, usada com o comando DRAW para criar formas e desenhos complexos.

Pode-se "pegar" qualquer retângulo no vídeo com um comando GET, armazená-lo numa matriz em BASIC, e substituí-lo na tela com um PUT. Isso facilita desenhar padrões geométricos e produzir animação simples e efeitos especiais, como a inversão de uma imagem. O toque final está nos sprites. O chip 9929 do monitor permite ao usuário desenhar suas próprias formas animadas: pessoas, invasores do espaço, mísseis.

A instrução ON SPRITE GOSUB permite criar uma "rotina de colisão": o programa roda normalmente, mas, se dois sprites colidem, desvia-se para essa rotina, que pode ser usada, por exemplo, para detectar o choque entre um míssil e uma astronave. Dessa forma, o programador não precisa checar continuamente todos os eventos como possibilidade de ocorrer; e os programas ficam mais fáceis de elaborar e rodam muito mais depressa. Rotinas como essa também estão disponíveis para as teclas de funções e para as que movem o cursor, entre outras.

Um bom som é essencial, e o chip sonoro do 318 possui três vozes e toda uma gama de efeitos especiais. O som pode ser reproduzido por televisão, o que permite um adequado controle de volume.

Há no 318 uma seleção de interfaces: duas saídas para joystick, uma entrada para cartucho, uma para cassete e um conector de expansão. Existe no mercado uma variedade de acessórios interessantes, mas costumam ser caros. Para conectar um deles, necessita-se de um miniexpansor, que permite memória extra, em geral uma expansão de 16 ou 64 Kbytes. Uma expansão ainda maior é conseguida mediante o superexpansor, que pode conectar até sete acessórios, utilizando um sistema de slots semelhante ao do Apple II. Incorporam-se mais memória, interfaces para impressora, unidades de discos e modems.

Se você for um entusiasta por jogos, poderá optar por um acessório muito interessante, o adaptador de jogos Coleco, que possibilita ao Spectravideo rodar cartuchos de videojogos dessa marca. Mas esse é um meio comparativamente dispendioso de rodar software de jogos.

O 318 requer um gravador cassete de sua própria marca, característica encontrada em núme-

Joystick do Spectravideo
Substitui as usuais teclas de movimentação do cursor. Sem o cabo, pode-se tocar um dos lados do disco para subir ou descer o cursor, ou movê-lo para a direita ou para a esquerda. O joystick tem o mesmo efeito, mas, empurrado em duas direções ao mesmo tempo, possibilita o movimento em diagonal, recurso impossível com as quatro teclas tradicionais.

Entrada para cartucho
Os cartuchos se encaixam neste conector, firmemente montado na placa principal.

ROM
O BASIC reside em duas ROMs de 16 Kbytes.

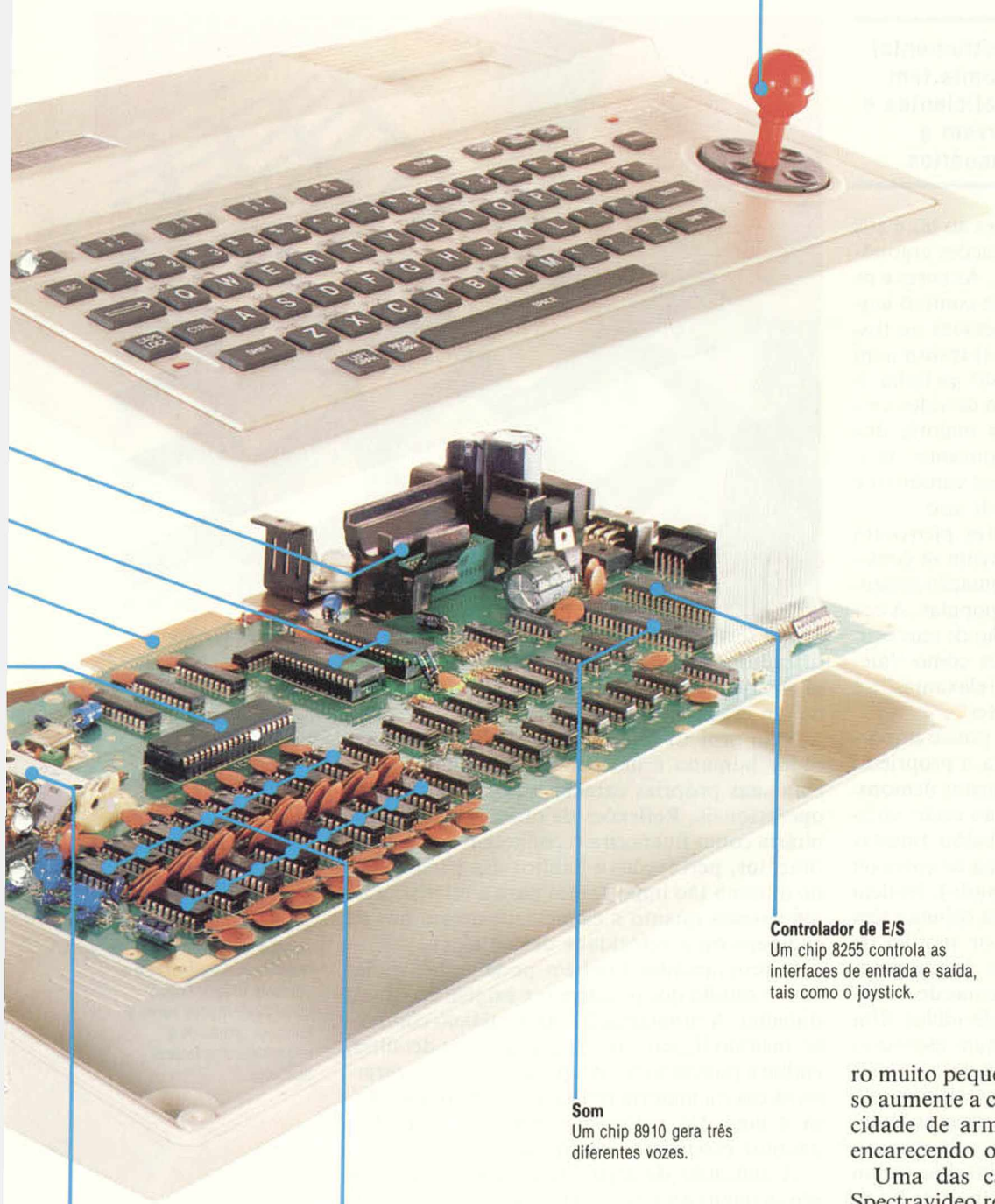
Conexão de expansões
Conectam-se aqui diversas expanções opcionais.

CPU
O processador utilizado é o conhecido Z80.

Saída para monitor
Acopla um modulador separado para acionar um aparelho de tevê. Isso permite usar o 318 com televisores de diferentes padrões, bastando selecionar o modulador apropriado.

Saída para cassete
Interface para gravador cassete de uso exclusivo do Spectravideo.

Versão aprimorada
Uma atraente alternativa ao 318 é o 328, a versão mais sofisticada, com teclado móvel, 80 Kbytes de RAM e software processador de texto residente. Tem por objetivo iniciar os que pretendem familiarizar-se com equipamentos profissionais.



Joystick incorporado
Um disco com cabo de joystick substitui as convencionais teclas do cursor.

Controlador de E/S
Um chip 8255 controla as interfaces de entrada e saída, tais como o joystick.

Som
Um chip 8910 gera três diferentes vozes.

Vídeo
O chip de vídeo 9929, encapsulado num dissipador térmico.

RAM
Dezesseis chips fornecem 32 Kbytes de RAM.

SPECTRAVIDEO 318

CPU

Z80.

MEMÓRIA

32 Kbytes de RAM, dos quais cerca de 12 Kbytes disponíveis para programas em BASIC; 32 Kbytes de ROM.

TELA

Modo texto: 24 linhas x 40 colunas. Modo gráfico: alta resolução, 256 x 92 pixels, dezesseis cores e recursos para desenhar sprites. Há também uma opção de 80 colunas.

INTERFACES

Conector de expansões, entrada para cartucho, duas saídas para joystick.

TECLADO

Com atenuador de pressão; teclas de função e de edição; joystick incorporado, para movimento do cursor.

LINGUAGENS DISPONÍVEIS

BASIC.

DOCUMENTAÇÃO

Reduzida e falha, mas a chegada de outros micros MSX deve encorajar publicações independentes.

ro muito pequeno de computadores. Embora isso aumente a confiabilidade do sistema e a velocidade de armazenamento e de leitura, acaba encarecendo o equipamento.

Uma das características mais atraentes do Spectravideo reside em sua capacidade de expansão. Com a adição do superexpansor para mais 64 Kbytes de RAM, placa de 80 colunas e uma unidade de discos, a máquina se converte num pequeno computador comercial CP/M, com acesso a software profissional. Mas ainda é difícil encontrar programas compatíveis com seu formato de disquete, problema que persistirá até que o equipamento deixe de ser uma novidade e se popularize.

SOB MEDIDA

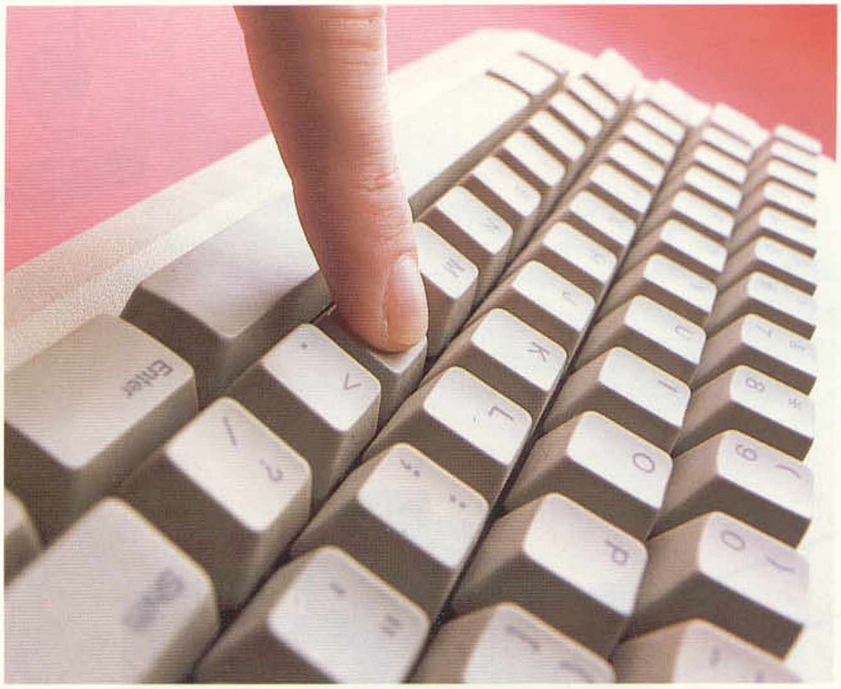
Adaptando o ambiente e o instrumental ao trabalho humano, a ergonomia tem por objeto projetar sistemas eficientes e que, ao mesmo tempo, preservem a saúde e o conforto de seus usuários.

A chegada dos microcomputadores ao lar e aos locais de trabalho torna as implicações ergonômicas importantes para todos nós. As cores e os sons constituem bons exemplos de como o ambiente afeta o desempenho das pessoas no trabalho. Algumas centrais telefônicas transmitem música para quem está aguardando na linha, a fim de amenizar a espera. Uma tela de vídeo verde ou âmbar é considerada pela maioria dos usuários agradável à vista e repousante. Já o branco e preto convencional parece cansativo e desgastante após algumas horas de uso.

Diferentes combinações de cores provocam reações bastante diversas: geralmente se considera azul sobre amarelo uma combinação atraente, enquanto azul sobre verde é impopular. A cor de uma sala pode afetar a disposição de seus ocupantes: tons amarelados são tidos como "alegres"; o azul e o verde, como "relaxantes"; e o marrom e o cinza, como "tristes".

Tais diferenças podem parecer pouco importantes ou mesmo irrelevantes para o proprietário de um micro. Mas os ergonomistas demonstraram que as pessoas, quando não estão satisfeitas com seu ambiente de trabalho (muitas vezes sem saberem que é o esquema de cores ou o ruído de fundo que as está afetando), tendem a sentir mais fadiga ocular, dores na coluna e têm maior probabilidade de faltar por motivo de doença. É muito comum usuários de computadores sofrerem por causa de esquemas de trabalho mal estruturados e salas mal divididas. Um importante estudo ergonômico, num escritório em que os operadores trabalhavam em terminais dispostos junto às paredes da sala, relacionou a baixa eficiência e o alto índice de erros ao isolamento social e desgaste causados pela disposição da sala. Quando os funcionários foram colocados frente a frente em terminais de baixa altura dispostos no meio da sala, o ambiente ficou mais agradável e a qualidade do trabalho melhorou sensivelmente.

A adoção de computadores resultou em menor especialização em alguns empregos, tornando o trabalho das pessoas desestimulante, rotineiro e pouco interessante. Hoje se reconhece como parte do trabalho do analista de sistemas garantir que as tarefas atribuídas às pessoas após a instalação de um computador sejam gratificantes e suficientemente estimulante. E, nessa análise, é necessária a orientação especializada de um ergonomista.



As teclas do sucesso
A interação do usuário com o computador começa pelo teclado, que já foi submetido a muitos aperfeiçoamentos ergonômicos — desde teclas côncavas e teclados destacados, inclináveis, até visores de cristal líquido. Mas seu maior defeito — a ineficiência da disposição tipo QWERTY — parece insolucionável, pois a maioria das pessoas reluta em aprender a datilografar segundo novos padrões. Tais sentimentos de resistência a mudanças e o aparente irracionalismo constituem muitas vezes o principal obstáculo à engenharia dos fatores humanos.

Num projeto de sistema ergonômico, o operador humano é um componente importante, com suas próprias características e tolerâncias operacionais. Reflexões de ordem tanto humanitária como financeira reconhecem que os sentimentos, percepções e hábitos das pessoas são no mínimo tão importantes para a eficiência de um sistema quanto a capacidade de seu bus de endereço ou a velocidade de seu clock.

O programador também pode se beneficiar com o estudo dos problemas e exigências de seu trabalho. A programação exige cuidado constante, método lógico e muita atenção aos detalhes, embora poucas pessoas apresentem essas características e a maioria resista à sua imposição. Essa é uma das razões por que não se podem garantir programas isentos de erro.

A aplicação da ergonomia na elaboração de novas linguagens de programação e métodos de desenvolvimento de sistemas é uma técnica fascinante, oferecendo muitas vantagens aos programadores e seus empregadores. Os psicólogos já dedicaram anos ao estudo das pessoas enquanto sistemas de processamento de informações, tendo hoje idéias claras (embora ainda incompletas) sobre estruturas da memória, velocidade e habilidade cerebral. Essas informações podem auxiliar no desenvolvimento de controles de pro-

gramas e na estruturação de dados com os quais as pessoas se sintam à vontade e que possam usar naturalmente ou com pouco treinamento.

Considerações de ordem psicológica há muito foram incorporadas aos métodos de treinamento, seleção e organização industrial. De modo ainda mais significativo, os psicólogos podem garantir que os projetistas se concentrem primeiramente em fazer os sistemas adequados às pessoas, e não o contrário: supor que o usuário irá se adaptar ao sistema.

Nos últimos anos, os ergonomistas vêm estudando a maneira como as pessoas reagem aos softwares complexos — a chamada "interface do usuário". No início da computação, os usuários eram profissionais bem treinados, altamente motivados, preparados para aceitar desconforto e inconveniências, dispostos a atingir um nível de competência técnica proporcional à potência do equipamento. Hoje o usuário é literalmente um leigo, uma pessoa comum, pouco tolerante com máquinas temperamentais ou exigentes. Se tivesse de aprender uma linguagem de informática para utilizar o caixa eletrônico de um banco, ele provavelmente encarregaria alguma instituição financeira de cuidar de seus negócios. E não são apenas os usuários ocasionais que têm problemas de interface com as máquinas. Os sistemas de bancos de dados que colocam megabytes de informações em cada mesa de escritório são, em toda parte, subutilizados ou utilizados erradamente; extrair deles informações exige fluência em linguagens computacionais complexas. Pesquisadores e usuários esperam que a próxima geração de sistemas aplicativos torne possível a comunicação com o banco de dados ou o modelo financeiro em linguagem próxima à comum. Nesse tipo de sistema, um primeiro nível do terminal traduzirá a entrada do usuário para a linguagem de comando do sistema, pedindo-lhe que especifique melhor suas solicitações e explicando ou expandindo as respostas do sistema.

Há muitos modos de auxiliar o usuário, inclusive recursos de "auxílio em linha", incorporados aos softwares mais avançados. A pesquisa no campo da inteligência artificial levou ao desenvolvimento de programas especialistas, que reúnem grande número de conhecimentos e podem explicar o processo pelo qual se chega a decisões. Acredita-se que essas técnicas ajudarão a criar "módulos de orientação" para auxiliar os usuários. Esse tipo de software "inteligente" dá, entretanto, origem a uma questão filosófica: o que é uma boa explicação e para quem se dirige? O programador poderia querer uma explicação das estruturas dos dados e dos processos de um sistema; já o usuário financeiro preferiria um comentário sobre os meios de se atingir um objetivo comercial. Tais diferenças são problemas de semântica e de lingüística para os ergonomistas e os cientistas de computação.

Outra área nebulosa no estudo da interface do usuário é conhecida pelos psicólogos como "modelagem". As pessoas utilizam modelos mentais

(por exemplo, estereótipos nacionais ou raciais) para preencher lacunas em seu conhecimento; muitos de nós emitiríamos, sem titubear, opiniões sobre a aparência, pontos de vista, personalidade e posições políticas de um estranho simplesmente e a partir do conhecimento de seu trabalho — funcionário público, cobrador de ônibus, maestro. De modo semelhante, as pessoas abordam os computadores com idéias estereotipadas sobre o poder e comportamento das máquinas, podendo imaginá-las mais inteligentes e com maior conhecimento do que os seres humanos no desempenho do mesmo tipo de tarefa. Quando encontram respostas impessoais e lacônicas dadas por muitos programas de softwares (especialmente em relação a entradas incorretas), elas podem considerar os computadores mal-educados, inamistosos e até mesmos hostis — julgamentos totalmente impróprios para máquinas. Essa percepção personalizada da máquina afeta toda a interação, muitas vezes levando a respostas inadequadas de ambos os lados do console.

Programadores que tentam incentivar a interação podem agravar o problema, pela introdução de características que fazem o programa parecer mais inteligente do que realmente é. Por exemplo, mensagens amistosas como "Olá, João, eu sou o espírito do banco de dados" podem incentivar o usuário a responder no mesmo estilo, muitas vezes com resultados catastróficos e correspondente desestímulo.

A interação requer uma abordagem mais hábil, envolvendo reflexão e preocupação com as pessoas, bem como admissão de suas reações complexas aos computadores e ao trabalho. Ergonomia, portanto, significa bem mais que inclinar o vídeo ou pintar o computador de lilás.

Condições de trabalho
De modo geral, admite-se que as pessoas no trabalho têm seu desempenho afetado por elementos físicos fáceis de notar, como a altura da mesa de trabalho, a temperatura ambiente e os níveis de ruído. Efeitos mais sutis, entretanto, como cores, luzes e percepção espacial, estão sendo reconhecidos agora como aspectos igualmente importantes no local de trabalho, em especial onde há computadores. Ao instalar um novo sistema, o analista deve considerar todos esses fatores, além de escolher o hardware e o software.

BASE LUNAR

Este programa auto-explicativo permite explorar toda a capacidade gráfica do TK 90X. Ao mesmo tempo, é um fascinante jogo de ação, que se apresenta em dois níveis de dificuldade.

Você tem por incumbência defender uma base lunar, sujeita a freqüentes ataques de naves alienígenas que tentam destruir suas reservas de combustível. Para tanto, dispõe de um canhão laser, cuja mira aparece na tela. O movimento do canhão é comandado pelas teclas [5], [6], [7] e [8]: aponte para o inimigo e dispare com o [0].

A tática usada pelos adversários é a de aproximar-se ao máximo dos tanques de combustível, a fim de não falharem no ataque e de tentar destruí-los completamente. Assim, quanto menor for a distância do inimigo abatido, mais pontos você ganhará. Mas cuidado, pois o menor erro nesse instante crítico pode custar caro. Você pode optar entre dois níveis de jogo: [0], difícil; e [1], fácil.

O algoritmo do programa foi construído utilizando lógica modular; assim, por meio do GO-SUB, o programa principal gerencia várias sub-rotinas, o que possibilita a observação de trechos isolados do mesmo.

Um efeito que pode ser indesejável é a lentidão do programa. Contornamos isso facilmente eliminando alguns comandos de som e sub-rotinas ou alterando o algoritmo para lógica estruturada. Outro recurso consiste no emprego de linguagem de máquina.

O programa que segue já inclui uma rotina geradora de caracteres nas linhas 750, 760, 780, 786 e 790. Se você optar por redefinir os caracteres, basta omitir essas linhas ao digitar e seguir as instruções de redefinição. Do contrário, deve mudar os caracteres criados pelos modos gráficos das letras A, B, C, D, E, F, e G da seguinte forma:

Posição	Caracteres usados
Afastada	B
Média	C, D
Próxima	C, E, D
Mira	A
Explosão	F, G

Redefinição de caracteres

Os caracteres gráficos especiais precisam ser definidos antes da digitação do programa. O TK 90X dispõe de um editor de caracteres, acessado pela função UDG 2, o que você deve fazer logo depois de ligar o computador.

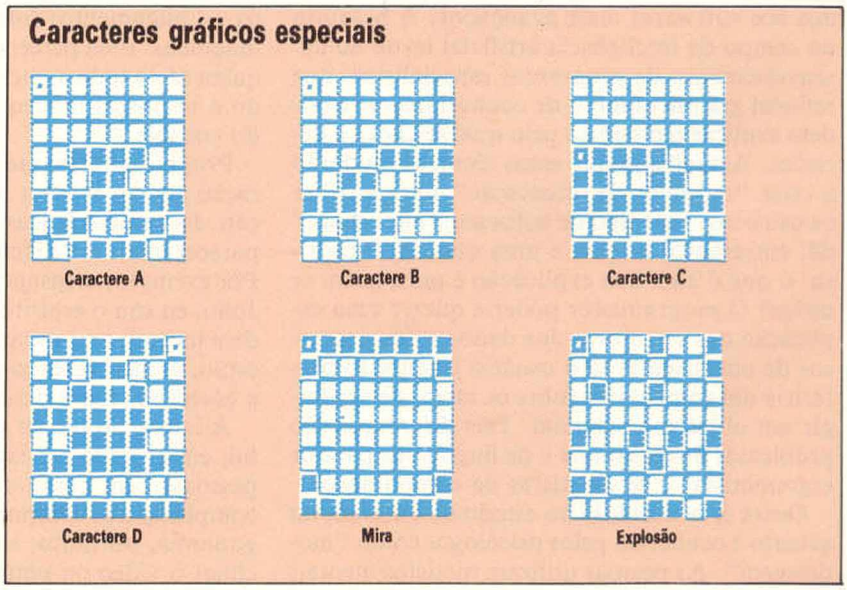
Quando se acessa a função, surgem na tela as letras do alfabeto, de A até U, representadas numa matriz 8 x 8. Para criar os caracteres, digite a letra a ser desenhada; pressione em seguida as teclas [Caps Shift] ou [Simbol Shift] e a [1], simultaneamente, para limpar a matriz. Esse procedimento faz aparecer um ponto no canto superior esquerdo, que pode ser deslocado em qualquer direção no interior da matriz, por meio das teclas [5], [6], [7] e [8]. É com esse ponto que se formam os caracteres. Para tanto, escureça os quadros em que ele aparece, com uma das teclas de movimento pressionada junto com a [Caps Shift].

Apague o ponto também com uma das teclas de movimento mais a [Simbol Shift], ambas pressionadas ao mesmo tempo.

Para criar o caractere seguinte, digite a tecla correspondente, mas sem pressionar [Caps Shift]. Esta, acionada simultaneamente com [0], permite sair do modo de edição de caracteres; e, com a [9], possibilita acessar os caracteres gerados e obter o cursor G. Digitando a tecla correspondente, você obtém o símbolo gráfico criado no modo UDG 2.

EFEITO DE APROXIMAÇÃO	
Posição	**Caracteres usados**
Afastada	A
Média	B, C
Próxima	B, D, C

Caracteres gráficos especiais



```
  1 REM BASE LUNAR
 20 GOSUB 900
 30 GOSUB 700
 35 GOSUB 800
 40 GOSUB 300
 45 GOSUB 100
 50 IF ht<>1 THEN GOSUB 100
 70 PRINT INK 5; OVER 0;AT 19,6
;po;" ";AT 19,25;sc;AT 20,6;lz;A
T 20,15;tno;AT 20,25;ki
 80 IF po<0 THEN GOTO 600
 90 GOTO 40
100 LET uxo=ux: LET uyo=u
101 LET u$=INKEY$
105 IF u$="" THEN RETURN
106 IF u$="0" OR u$="1" THEN GO
TO 150
107 IF u$<"5" OR u$>"8" THEN RE
TURN
110 LET ux=ux+(u$="8" AND ux<29
)-(u$="5" AND ux>2)
120 LET uy=uy+(u$="6" AND uy<13
)-(u$="7" AND uy<13)-(u$="7" AND
 uy>2)
125 IF ux=uxo AND uy=uyo THEN R
ETURN
130 PRINT INK 7;AT uyo,uxo;"□";
 INK 6;AT uy,ux;"□": LET po=po-1
 SOUND .005,20-uy: RETURN
150 INK 5: LET po=po-10
155 LET lz=lz+1: LET lcx=ux*8+4
 LET lcy=(21-uy)*8+4
160 FOR n=1 TO 2: SOUND .04,6:
SOUND .02,12: PLOT 8,56: DRAW lc
x-8,lcy-56: PLOT 247,56: DRAW lc
x-247,lcy-56: INK 7: NEXT n
175 INK 6: FOR n=1 TO 2: PRINT
AT uy,ux-1;"▒▒▒": SOUND .03,24:
INK 7: NEXT n
180 IF NOT (uy=ty AND ux>=tx-df
 AND ux<tx+df+LEN t$) THEN INK 5
: RETURN
200 OVER 1: LET ht=0: LET ki=ki
+1: LET sc=sc+100-LEN t$*10
210 FOR n=1 TO 4: PRINT AT uy-1
,ux-1;"\*/";AT uy,ux-1;"▒▒▒";AT
uy+1,uy-1;"/*\": SOUND .1,36: SO
UND .05,6: INK 7: NEXT n: LET ht
=1
220 RETURN
300 IF ht=1 THEN GOTO 360
302 LET tyo=ty: LET txo=tx
305 LET u$=t$: LET tx=tx+1
310 IF ty>=12 THEN LET tc=-1
311 IF ty<=4 THEN LET tc=1
312 LET ty=ty+tc+(INT (RND*2) A
ND ty<14)-(INT (RND*2) AND ty>3)
315 LET t$="": LET t$=("*" AND
tx<9)+("▒" AND tx>=9 AND tx<18)
+("▒" AND tx>=18)
330 PRINT INK 4;AT tyo,txo;u$;A
T ty,tx;t$: SOUND .02,-12
343 IF ty>8 AND RND<.7 AND ((tx
>16 AND tx<21) OR tx=24) THEN IN
K 2: FOR m=1 TO 2: FOR n=-1 TO 2
: PLOT tx*8+12,175-(ty*8)-12: DR
AW n*3,-40: SOUND .003,30: NEXT
n: INK 7: PRINT INK 6;AT 21,0;"▒
▒▒▒
```

SALTOS E DESVIOS

A linguagem ASSEMBLY utiliza flags e instruções de salto relativo para direcionar o fluxo de controle do programa. Veremos aqui como esses recursos são combinados para criar loops e tabelas de dados.

Para que possamos utilizar os vários modos de endereçamento da CPU, especialmente os endereços indexados, é necessário em primeiro lugar construir loops. Sem essa estrutura básica, ficaríamos na situação de um programador em BASIC que sabe armar matrizes, mas não conhece o comando FOR-NEXT. Não há na linguagem ASSEMBLY uma estrutura automática de loop como o FOR-NEXT (embora o Z80 possua uma instrução bem semelhante), mas podemos criar loops do tipo IF-THEN-GOTO. Estes requerem instruções que tomam decisões ou expressam condições, e irão modificar a ordem de execução das instruções do programa.

A tomada de decisões na linguagem ASSEMBLY baseia-se nos flags do registrador de estado do processador. Eles mostram o efeito da última condição executada sobre o acumulador; por isso são às vezes chamados de flags de condição. Todos atuam na tomada de decisões, mas consideraremos apenas dois deles — o flag zero (Z) e o de transporte (C, "carry").

O estado desses flags determina se o processador irá executar a próxima instrução do programa ou se irá saltar para alguma outra em qualquer ponto do programa. Para que o processador decida se irá continuar na seqüência ou saltar para outra instrução, ele deve aceitar (ou então modificar) o endereço existente no contador do programa. Esse registrador sempre contém o endereço do próximo comando em código de máquina a ser obedecido. Ao começar a executar uma instrução, o processador carrega o código de operação dessa ordem. O endereço contido no registrador é incrementado de acordo com o número de bytes da instrução, de modo que o contador do programa passa a indicar o código de operação seguinte. Se a instrução atual fizer o contador do programa indicar um endereço em outro ponto, então realiza-se um salto, ou desvio (abreviado por J, de "jump", ou B, de "branch").

No processador 6502, o comando BEQ altera o contador do programa se o flag zero está ligado; e BCS fará o mesmo se o flag de transporte estiver ativado. No Z80, esses comandos são JR Z e JR C, respectivamente. Esses quatro códigos de operação chamam-se instrução de desvio, pois representam um ponto de desvio no fluxo de controle do programa. Seu operando é um número de um só byte, que, acrescentado ao endereço do contador, produz um novo endereço. Veja o que acontece na execução desse programa.

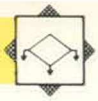
ORG $5E00			
6502		Z80	
5E00	ADC #$34	ADC	A,$34
5E02	BEQ $03	JR	Z,$03
5E04	STA $5E20	LD	($5E20),A
5E07	RTS	RET	

Se a instrução ADC em $5E00 produzir um resultado zero no acumulador (improvável, mas pode acontecer), as instruções BEQ ou JR Z em $5E02 acrescentarão $03 ao conteúdo do contador do programa. Portanto, a próxima ordem a ser executada será o retorno a $5E07, saltando a instrução em $5E04.

À primeira vista, esse salto parece incorreto. Poderíamos pensar que, se a instrução em $5E02 acrescenta $03 ao contador do programa, o endereço nele armazenado deverá tornar-se $5E05. Mas é importante lembrar que o contador do programa sempre indica a próxima instrução a ser obedecida, e não aquela em execução no momento. Dessa forma, no início da execução da instrução em $5E02, o contador do programa conterá o endereço $5E04, que é a posição do comando seguinte. Quando se acrescenta $03 a $5E04, o resultado será $5E07, o endereço da próxima instrução.

É importante assinalar que o processador não tem a capacidade de checar se os endereços indicados são corretos. Se por engano determinarmos que o deslocamento será de $02, o contador do programa, contendo zero, será incrementado de $5E06 como o endereço do código de operação da instrução seguinte. Em nosso programa correto, $5E06 contém o valor $5E, que é o byte mais alto do operando da instrução em $5E04. Contudo, o processador não pode avaliar se esse valor é ou não o comando correto. Ele considera $5E como um código de operação válido, e passará a executá-lo, supondo os bytes após $5E06 como os operandos da instrução. O resultado será a falha total do programa. Esse tipo de deslocamento mal calculado constitui um dos erros mais freqüentes na programação em código de máquina.

Ao programar em linguagem ASSEMBLY, porém, o cálculo dos deslocamentos deixa de ser um problema porque quem o executa é o programa montador, ou compilador. Assim, em vez de determinar um deslocamento hexadecimal co-



mo operando da instrução de desvio, fornecemos apenas um endereço simbólico da instrução a ser executada. Isso facilita muito a compreensão dos programas em linguagem ASSEMBLY. O programa assemblador transforma o endereço simbólico num endereço absoluto, calcula o deslocamento necessário para atingir tal endereço e insere esse deslocamento na instrução em código de máquina. O endereço simbólico chama-se rótulo e equivale a um número de linha nos programas em BASIC.

Examinemos mais detalhadamente o uso dos rótulos. Um rótulo é um string alfanumérico colocado no início de uma instrução em linguagem ASSEMBLY. O compilador o considera como um símbolo de 2 bytes, representando o endereço do primeiro byte da instrução. Assim, podemos reescrever nosso programa utilizando o rótulo EXIT para representar o ponto de saída:

ORG $5E00		
	6502	Z80
5E00	ADC #$34	ADC A,$34
5E02	BEQ EXIT	JR Z,EXIT
5E04	STA $5E20	LD ($5E20),A
5E07 EXIT	RTS	RET

A instrução em $5E02 significa agora "IF o valor do acumulador for zero THEN GOTO o endereço representado por EXIT". Essa versão do programa fica mais compreensível que a anterior, diminuindo a possibilidade de cálculo errado quanto ao ponto de destino do salto.

Podemos agora criar um loop com as instruções de desvio e os rótulos EXIT e START, indicando este último o ponto inicial:

ORG $5E00		
	6502	Z80
5E00 START	ADC #$34	ADC A,$34
5E02	BNE START	JR NZ,START
5E04	STA $5E20	LD ($5E20),A
5E07 EXIT	RTS	RET

Observe a utilização das novas instruções de desvio BNE e JR NZ, ambas significando: "Desviar se o acumulador não for igual a zero". Vejamos o efeito desse programa. A primeira instrução acrescenta $34 ao acumulador. Se o resultado não for igual a zero, o programa desviará, retornando a $5E00, que é o endereço representado por START. Mais uma vez, acrescenta-se $34 ao acumulador e, conforme o resultado, ocorrerá novo desvio. Esse loop continuará repetindo a instrução ADC, até que se satisfaça a condição de desvio. Quando o conteúdo do acumulador for igual a zero, não ocorrerá o desvio em $5E02; em vez disso, será executada a instrução em $5E04.

Isso equivale a um loop IF-THEN-GOTO em BASIC; contudo, fica difícil compreender como o acumulador pode chegar a zero, pois ele é incrementado em $34 a cada execução do loop. Como poderá totalizar zero? O acumulador é um registrador de apenas um byte. Se a adição resulta num número de 2 bytes, ativa-se o flag de transporte do registrador de estado do processador; assim, o acumulador conterá o byte mais baixo do resultado. Se o acumulador contiver $CC, por exemplo, o acréscimo de $34 resultará em $0100, um número de 2 bytes. O flag de transporte será ativado, e o acumulador conterá o byte mais baixo desse resultado — $00. Assim, o conteúdo do acumulador se tornará zero e, como resultado, o flag zero será ativado.

Tendo em mente esse resultado, poderíamos reescrever o programa usando outra condição de desvio, incorporando o estado do flag de transporte, e não do flag zero:

ORG $5E00		
	6502	Z80
5E00 START	ADC #$34	ADC A,$34
5E02	BCC START	JR NC,START
5E04	STA $5E20	LD ($5E20),A
5E07 EXIT	RTS	RET

Nessa versão, a instrução em $5E02 significa: "Se o flag de transporte estiver desligado, desviar para START". Quando o acréscimo de $34 do acumulador resultar num valor maior que $FF, o flag de transporte será ativado, e não haverá retorno ao endereço START.

Contadores de loop

O desvio de acordo com a condição dos flags de transporte e zero parece ser um recurso bastante limitado; permite, contudo, inúmeras possibilidades de tomadas de decisão. O que nos falta agora é um contador de loop, para contar, por exemplo, o número de execuções de um loop ou para determinar a saída dele após um certo número de iterações. Atinge-se o primeiro objetivo facilmente, pelo emprego de um registrador de índice da CPU para manter o contador e de uma instrução de incremento para atualizá-lo:

6502		
0000	ORG	$5DFD
5DFD	LDX	#$00
5DFF START	INX	
5E00	ADC	#$34
5E02	BCC	START
5E04	STX	$5E20
5E07 EXIT	RTS	

Z80		
0000	ORG	$5DFA
5DFA	LD	IX,$0000
5DFE START	INC	IX
5E00	ADC	A,$34
5E02	JR	NC,START
5E04	LD	($5E20),IX
5E08 EXIT	RET	

A nova estrutura causou várias alterações no programa. Em primeiro lugar, as instruções inseridas no início exigem um novo endereço ORG. Essas instruções têm efeito semelhante nos processadores 6502 e Z80, mas seus tamanhos são diferentes, de modo que os endereços de posição diferem nas duas versões do programa.

Em segundo lugar, usamos novas versões das instruções de carga (LDX, LD IX) e armazenamen-

to (STX, LD( ),IX) para colocar um valor inicial de $00 no registrador de índice da CPU. O registrador X do 6502 tem apenas 1 byte, mas o registrador IX do Z80 tem 2. Os registradores de índice têm funções especiais, mas são, em princípio, memória dinâmica da CPU, tal como o acumulador; aqui os utilizamos como acumuladores adicionais, para a contagem do loop. Na versão do 6502, quando ocorre a saída do loop, o conteúdo do registrador X é armazenado em $5E20. Na versão do Z80, o byte mais baixo do registrador IX é armazenado em $5E20, e o byte mais alto em $5E21.

Em terceiro lugar, uma nova instrução substitui a ADC como ponto inicial do loop: INX e INC IX são comandos de incremento, que acrescentam $01 ao conteúdo do registrador de índice. Isso atualiza o valor do contador de loops a cada iteração.

Podemos interpretar o programa assim: "Zerar o contador de loops, iniciar o loop incrementando o contador, acrescentar $34 ao acumulador e voltar para o início do loop se o flag de transporte estiver desligado. Do contrário, armazenar em $5E20 o conteúdo do contador de loops".

Uma modificação complementar aumentará muito a utilidade e abrangência do programa:

6502

```
0000          ORG   $5DFA
5DFA          LDX   #$00
5DFC START    STA   $5E22,X
5DFF          INX
5E00          ADC   #$34
5E02          BCC   START
5E04          STX   $5E20
5E07 EXIT     RTS
```

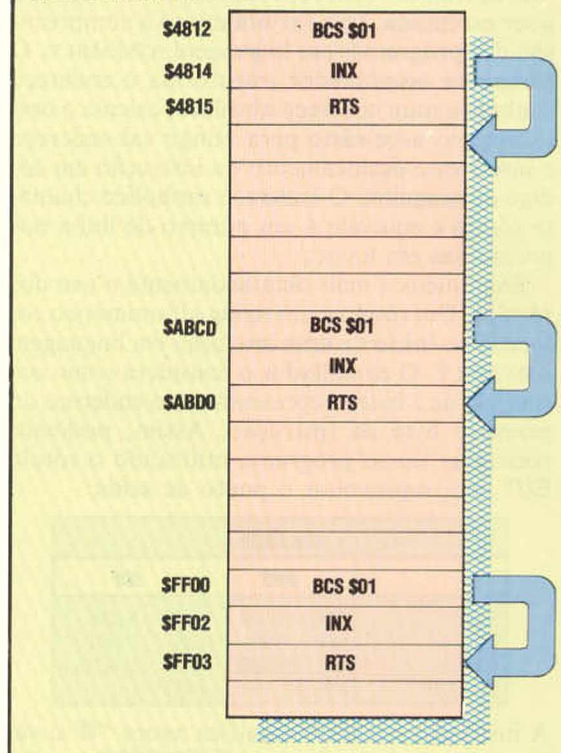
Z80

```
0000          ORG   $5DF7
5DF7          LD    IX,$5E00
5DFB START    LD    (IX+$22),A
5DFE          INC   IX
5E00          ADC   A,$34
5E02          JR    NC,START
5E04          LD    ($5E20),IX
5E08 EXIT     RET
```

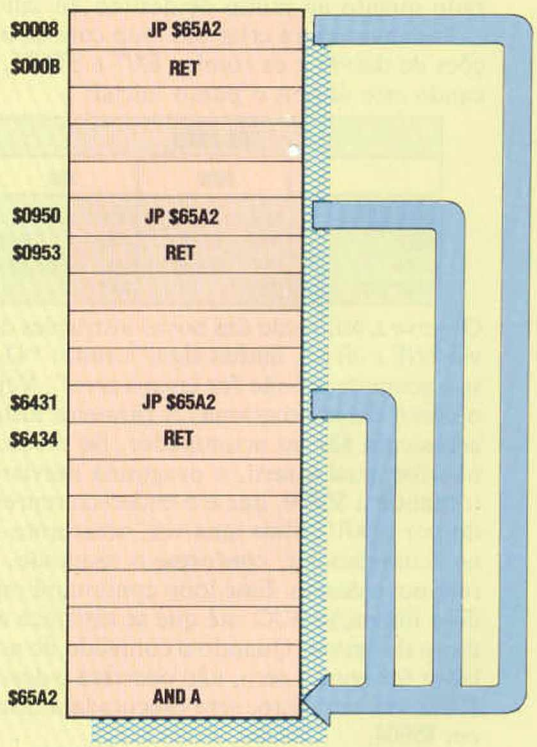
Ambas as versões do programa têm o mesmo efeito: criam, na posição $5E22, uma tabela de armazenamento dos valores sucessivos do acumulador, à medida que transcorre o programa, e armazenam em $5E20 o valor final do contador de loops, que equivale ao número de bytes da tabela.

A versão 6502 chega a esse resultado por meio da instrução STA $5E22,X, que significa "Acrescentar o conteúdo do registrador X ao endereço-base $5E22; então armazenar o conteúdo do acumulador do endereço assim formado". A instrução STA está aqui no modo indexado direto absoluto, isto é, ela usa o registrador X como um índice para modificar o endereço-base $5E22. Como o registrador X é inicializado em $00 e subseqüentemente incrementado a cada iteração, o valor inicial do acumulador será armazenado

SALTOS RELATIVOS



SALTOS ABSOLUTOS



em $5E22, o próximo valor em $5E23, e assim sucessivamente. Após a saída do loop, STX armazenará na posição $5E20 o valor final do contador de loops.

A versão do Z80 utiliza o registrador IX para indicar o endereço atual de armazenamento, empregando o byte mais baixo de IX como contador de loops. A instrução LD IX,$5E00 coloca no registrador IX o endereço-base $5E00; assim, o byte mais baixo de IX conterá $00. A instrução LD (IX + $22),A, de aparência um pouco estranha, significa "Acrescentar o endereço contido em IX e $22 e armazenar o conteúdo do acumulador no endereço assim formado". Como IX é inicializado em $5E00 e incrementado a cada iteração do loop, o valor inicial do acumulador será armazenado em $5E22, o seguinte em $5E23, e assim por diante. O byte mais baixo de IX vai registrando o número de iterações do loop, valor que é armazenado em $5E20, quando o loop se encerra. A instrução LD(IX + $22),A encontra-se aqui no modo de endereçamento indexado indireto absoluto, de complexidade um pouco maior que na versão do 6502, porém muito mais potente.

Agora já examinamos como a linguagem ASSEMBLY manipula loops e tabelas, técnicas de programação de grande utilidade. No próximo artigo veremos o uso prático de ambas.

Saltos relativos
A maioria das instruções de desvio, como BCS (que significa "Desviar se o flag de transporte estiver ligado") e JR NZ ("Desviar se o acumulador for não zero"), age de acordo com a condição do registrador de estado do processador, e utiliza o modo de salto relativo para redirecionar o fluxo de controle do programa.
A alternativa é o salto absoluto.

No exemplo, a instrução BCS $01 sempre causa um salto relativo de 1 byte para a frente (quando ocorre o salto, pois ela está condicionada ao estado do flag de transporte). Nesse salto, não importa em qual endereço se encontra o código de máquina. Aqui, a instrução BCS $01 é sempre seguida pela INX, que possui um só byte; portanto, quando se ativa o flag de transporte, BCS faz o programa saltar a instrução INX.

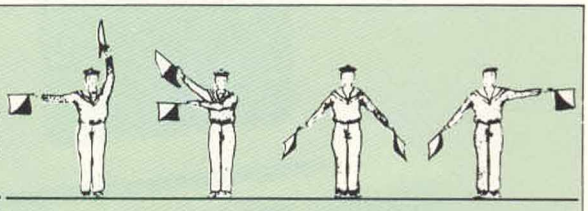
Saltos absolutos
Neste exemplo, a instrução JP $65A2 ocasiona um salto incondicional toda vez que aparece. Seu efeito é redirecionar a execução do programa para o endereço que forma seu operando — nesse caso, $65A2. Não se realiza qualquer teste, e o endereço da instrução em execução não é significativo, pois a execução do programa sempre continua a partir do endereço especificado.

Ambos os modos de salto apresentam vantagens e desvantagens, mas o critério mais importante na escolha entre um salto relativo e um absoluto está no reposicionamento. É bastante comum, na programação em ASSEMBLY, escrever-se uma rotina, montá-la num dado endereço ORG, e depois reutilizá-la com a mesma forma, mas com um valor ORG diferente. Se todos os saltos na rotina forem relativos, a modificação dos endereços das instruções não terá qualquer importância, e o programa fluirá sem dificuldade, seguindo as rotas desejadas; entretanto, se algum dos saltos for absoluto, quando se montar a rotina com outro ORG, os saltos continuarão a direcionar o controle para o endereço especificado, que pode então não ter qualquer significado para a rotina.
Os saltos relativos são reposicionáveis, mas os absolutos não.

Exercícios

Vimos neste artigo muitos pontos importantes — e tal vez complexos —, e somente a prática das novas instruções e modos de endereçamento permitira compreendê-los bem.

Use o programa compilador para montar e gravar as rotinas apresentadas neste artigo. Ao executar uma rotina, use o modo de depuração (debug) para examinar as posições de memória afetadas. É uma boa idéia inicializar sempre essas posições com uma constante identificável — $FF, por exemplo — antes da execução, de modo que posteriormente seja possível verificar se a memória foi alterada pelo programa. Para isso, use o comando debug Alter, ou mesmo o comando debug Move. Como sempre, lembre-se de que os endereços dados no programa são apenas exemplos; escolha, pois, endereços adequados à sua máquina.



Flag — levantando a bandeira
A palavra flag, que em inglês significa bandeira, indica a ocorrência de uma condição num programa, por analogia às bandeirinhas usadas em sinalização. Um flag nada mais é que um bit utilizado como sinal. No estado zero, ou desligado, representa uma bandeira abaixada; quando, porém, ocorre determinada condição na execução do programa, o flag passa a valer 1, ou seja, a bandeirinha levanta. Desvia-se, assim, o fluxo de comando do programa de acordo com o estado do flag.

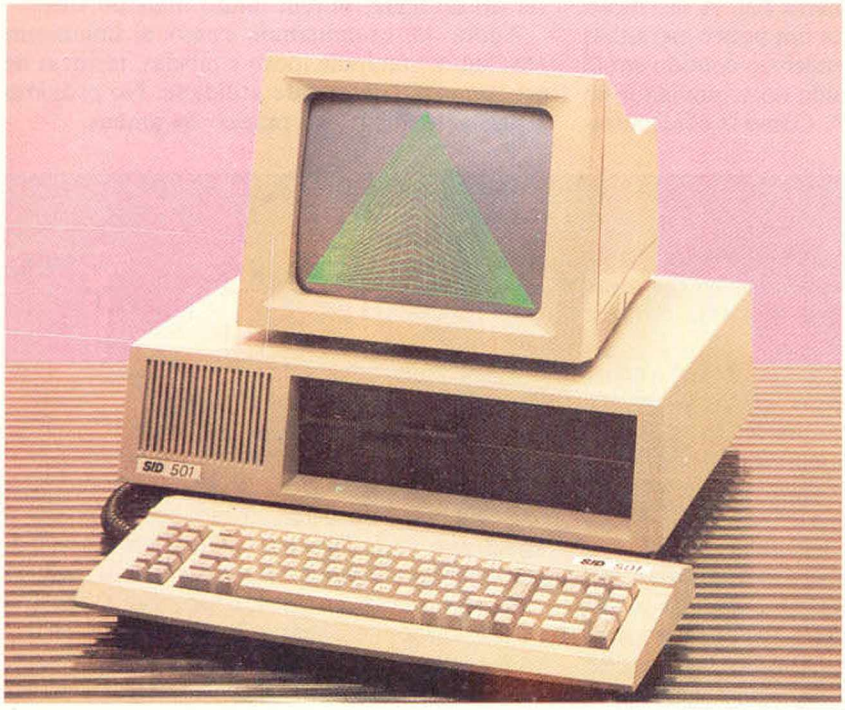
As instruções de desvio condicional, como vimos, dependem do conteúdo do PSR (Processador Status Register, "registrador de estado do processador"). Uma razão para se utilizar a opção de apresentação binária é possibilitar a verificação do conteúdo do PSR antes e depois da execução de uma instrução, e então observar as alterações nos flags. Não há, na linguagem ASSEMBLY uma única instrução para armazenar o conteúdo do PSR; assim, temos de usar os seguintes comandos:

Z80	
3E00 F5	PUSH AF
3E01 F5	PUSH AF
3E02 E1	POP HL
3E03 22 lo hi	LD (STORE1),HL
3E06 F1	POP AF
6502	
3E00 48	PHA
3E01 08	PHP
3E02 48	PHA
3E03 08	PHP
3E04 68	PLA
3E05 8D lo hi	STA STORE1
3E08 68	PLA
3E09 8D lo' hi'	STA (1+STORE1)
3E0C 28	PLP
3E0D 68	PLA

Essa seqüência de instruções armazenará o conteúdo atual do PSR no byte endereçado por STORE1 (que representa um endereço apropriado para sua máquina), enquanto o conteúdo do acumulador será armazenado em (1 + STORE1). Para usar essas instruções, basta inseri-las como um bloco, antes e depois da instrução cujo efeito você deseja observar. Entretanto, lembre-se de acrescentar 2 ao valor de STORE1 toda vez que inserir esse bloco. Após executar o programa, verifique a parte da memória onde se armazenaram os conteúdos do PSR e do acumulador.

Talvez lhe ocorra tratar esse bloco como uma subrotina, em vez de entrar com ele toda vez que necessário. Há no ASSEMBLY um equivalente do comando GOSUB do BASIC, mas seu uso aqui complicaria a programação, pois emprega a técnica do empilhamento. Ele iria interferir no uso que o bloco faz da mesma lista (PLA, PUSH, PHP etc. são manipulações de pilhas, que posteriormente explicaremos). Observe a diferença de tamanho entre a rotina do Z80 e a do 6502: isso se deve aos registradores de 2 bytes do Z80 e às instruções associadas.

SID



A SID Informática S.A. responde pela produção nacional — com tecnologia própria — de sistemas de automação bancária e comercial, de micro e minicomputadores.

Fundada em 1978, a SID Informática, empresa do grupo Sharp, é especializada na fabricação de micro e minicomputadores, bem como de sistemas de automação bancária e comercial, com 90% de tecnologia nacional.

Mas a história desse conglomerado de 43 empresas começa bem antes. Em 1961 foi fundado o conjunto de empresas Machline, que, em 1969, obteve a concessão para comercializar com exclusividade no Brasil os produtos da empresa japonesa Sharp Corporation. A partir daí, criou-se a Sharp Equipamentos Eletrônicos, que, apesar de possuir o mesmo nome da empresa japonesa, é totalmente nacional.

Quando instalou sua primeira unidade industrial na Zona Franca de Manaus, em 1972, a Sharp produzia calculadoras eletrônicas. Em 1974 foi uma das primeiras empresas a fabricar televisores em cores e, no ano seguinte, equipamentos de áudio. Hoje a Sharp Equipamentos Eletrônicos responde também por videocassetes e videocâmaras, empregando mais de 2.000 funcionários em suas quatro unidades industriais. Para aumentar a nacionalização de seus produtos, a empresa passou a investir na produção de componentes eletrônicos.

Na informática

Também nesse setor, a Sharp abarca parte considerável do mercado, através dos Sistemas de Informação Distribuída ou, simplesmente, SID Informática. Esta produz terminais financeiros, micro e minicomputadores e diversos outros equipamentos do setor de informática.

Para a automação bancária, a SID oferece uma linha de equipamentos que permite o serviço computadorizado de uma agência ou a integração de várias agências controladas por um computador central de grande porte. Esse núcleo de controle — o SID 3500 — processa todos os dados da agência e concentra as informações dos terminais de caixa, de retaguarda, de consulta e administrativos.

Na área comercial, a SID possui o terminal PDV (ponto de venda), que funciona como caixa registradora eletrônica e microcomputador.

O PDV SID-6000 tem processador de 16 bits e permite, entre outros serviços, a leitura de preços por código de barras, integração em redes de processamento para cadeia de lojas, leitura de cartão magnético para transferência automática de fundos, controle da contabilidade do estabelecimento e análise do giro do estoque.

Na área de microcomputadores, a SID comercializa três modelos: o 3000, o 3800 e o 3900. Todos equipados com microprocessador Intel 8085, memória MOS dinâmica de 64 Kbytes e ROM de 16 Kbytes. Possuem sistema operacional compatível com CP/M e interfaces que permitem ligação com outros micros, mínis ou computadores de grande porte, inclusive emulando terminais de outros fabricantes.

Os minicomputadores da série 5800 também apresentam três modelos, com capacidades variando entre 126 Kbytes e 1 Mbyte. Possuem processador central multiprogramado, e usam palavras de 16 bits.

Os mínis da SID destacam-se por utilizar o avançado banco de dados Adabas. Em 1985, a Sharp lançou o micro Hot-Bit, dos primeiros no Brasil a adotar o padrão MSX.

Em 1984, a SID assumiu o controle da fábrica de semicondutores da Philco, que passou a se chamar SID Microeletrônica. É a única empresa da América Latina que executa todas as fases de produção de semicondutores, componente fundamental para fabricação de equipamentos eletrônicos.

xadrez, com a maioria das casas alternadamente em preto e branco.

Algumas casas, entretanto, passam do preto para o branco, e de novo para o preto, à medida que o jogo prossegue. A razão disso é que as forças da Luz são mais poderosas nas casas brancas e as das Trevas, nas casas pretas.

O objetivo, no Archon, é ocupar os cinco "pontos de energia"; quatro deles situam-se nos vértices das bordas do tabuleiro e o principal, na casa central. Quando o jogo começa, a estratégia de abertura recomenda que o jogador retire seus ícones das casas de cor oposta à sua, onde estão vulneráveis ao ataque, movendo-os para aquelas em que são mais poderosos.

Ao fazer isso, o jogador abre passagem para a fileira de trás, o que possibilita o movimento de suas peças "de terra", permitindo-lhe montar um obstáculo efetivo de defesa contra as forças terrestres inimigas.

Nesse ponto, o jogo parece ter transposto a maioria dos elementos do xadrez para o computador. Mas a diferença real fica evidente quando se tenta tomar uma casa já ocupada por um ícone adversário. Ao mover, com o joystick, um ícone para uma casa já ocupada por um adversário, em vez de a peça inimiga ser automaticamente tomada, a tela apresenta uma ampliação da casa. As peças adversárias são posicionadas num dos lados da casa, em torno da qual há uma barra que representa a força do ícone, iniciando-se um tiroteio tipo fliperama.

Quando a barra é atingida pela peça adversária, a força diminui e, ao desaparecer por completo, os elementos opositores vencem e ocupam a casa. Como as forças e os métodos de ataque diferem radicalmente de um ícone para outro, algumas dessas batalhas são mais desiguais que outras. Por exemplo, um dragão pode, de sua casa, lançar fogo contra um oponente, mas um cavaleiro precisa estar muito próximo do inimigo para poder usar sua espada. As batalhas complicam-se mais ainda com barreiras espalhadas pelas casas, que mudam de cor à medida que o combate progride.

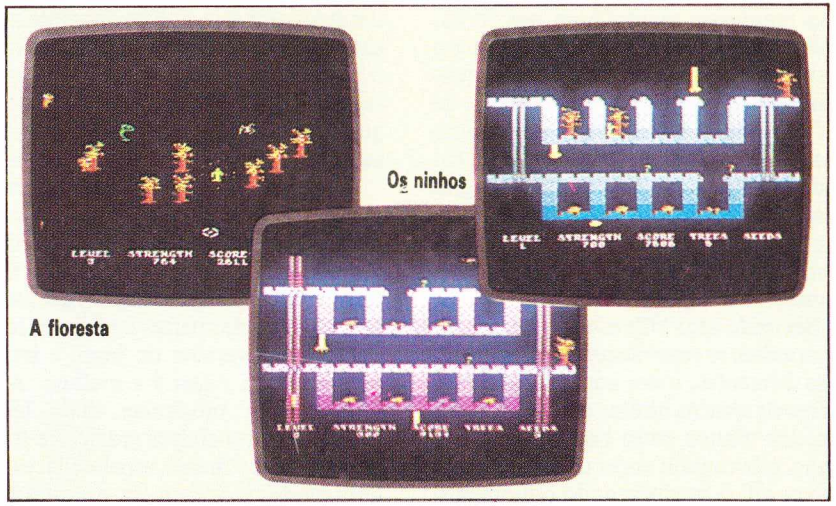
Cada lado tem uma figura capaz de utilizar a magia para combater: um mago para a Luz e um feiticeiro para as Trevas. Os encantamentos são idênticos, mas só podem ser usados pelo ícone uma única vez. As magias consistem em poderes tais como "ressuscitar" uma peça com maior movimentação para certa casa, ou a capacidade de convocar uma criatura chamada "elemental". Esta pode ser invocada para atacar uma peça inimiga e mostra-se útil quando um ícone adversário ameaça atacar uma peça mais fraca, ou quando se pretende enfraquecer o oponente. Entretanto, mesmo destruindo o inimigo, o elemental desaparecerá ao final da batalha, pois não pode ocupar uma casa.

O jogo é disputado contra o computador ou contra outra pessoa. Embora o computador não passe de um estrategista mediano, mostra-se um formidável oponente na categoria fliperama, e os principiantes podem se ver terrivelmente encurralados ao tentar vencer até mesmo uma simples batalha nas primeiras partidas.

Archon: Para os computadores Commodore 64, Atari e Apple II.
Editor: Ariolasoft, Inglaterra.
Autores: Anne Westfall, Jon Freeman e Paul Reiche III.
Joystick: Necessário.
Formato: Disco ou cassete.

MAGIA NEGRA

Necromancer (análise)



O Necromancer, da Synapse Software, pode ser comparado a uma peça teatral bem-feita. O roteiro se divide em três atos: as sementes do drama são literalmente plantadas no primeiro, desenvolvendo-se o enredo no segundo, até o clímax — o encontro entre o bem e o mal — no terceiro.

Abertura. As versões para o Atari e o Commodore 64 são carregadas com facilidade: o programa torna tolerável a lentidão da unidade de disco do Commodore, produzindo vários ruídos curiosos — quase como uma orquestra eletrônica em afinação. A página de abertura vem acompanhada por uma música introdutória, que usa a plenitude das possibilidades sonoras de cada máquina.

Primeiro ato: "A floresta". O drama começa na "idade das trevas", quando Tetragorn, o mago do mal, reina supremo. O jogador toma parte como Iluminar, um druida descrito como defensor da verdade e protetor da raça humana — coisa nada fácil, como bem se pode imaginar. O jogo tem início com o druida aparecendo num campo aberto e escuro. Uma aura de estrelas o protege das centenas de pequenos ogros que marcham implacavelmente pela tela, brandindo cutelos gigantes e tendo por fundo uma insistente trilha musical. O jogador emprega o joystick para controlar uma varinha mágica, que voa pelo vídeo. Essa é sua arma para destruir os ogros e marcar pontos. Depois, ela volta à mão do druida.

Festa visual
O Necromancer combina gráficos sprite com fundos em alta resolução, produzindo quadros deslumbrantes em três níveis de dificuldade. As brilhantes copas das árvores, a varinha mágica do druida, de ação rápida, o movimento das aranhas, os ogros, o fogo do Tetragorn, tudo isso confere ao Necromancer um tratamento visual estonteante, fazendo dele um jogo muito atraente.

Colocando a varinha mágica no espaço desejado e acionando o botão do joystick, o jogador pode plantar árvores, de modo a formar uma floresta, que depois o ajudará no jogo. Deve também procurar proteger as mudas contra os cutelos e as aranhas de Tetragorn; ao mesmo tempo, porém, precisa manter-se atento para sua força, que diminui quando o druida leva uma ferroada ou aumenta quando uma aranha morre. Depois de cinco níveis de dificuldade, o primeiro ato termina no ataque das aranhas, o qual esgota rapidamente a força do druida. O programa então congela a ação, conta o número de árvores que cresceram e transporta o mago do bem para o ato seguinte.

Segundo ato: "Os ninhos". É aí que as aranhas se reproduzem. Há cinco níveis diferentes, todos contendo dois andares de quatro ninhos cada um. Dentro dos ninhos estão os ovos das aranhas, que tomam cores diferentes conforme vai se aproximando o momento da eclosão. Também na tela aparece o "depósito de árvores" (que contém as árvores que cresceram no primeiro ato). O jogador deve usar sua varinha a fim de soltar e movimentar uma árvore para uma posição acima de um dos ninhos; se for rápido, as raízes da árvore crescerão, destruindo o ninho e impedindo que os ovos eclodam.

As "mãos do destino" acrescentam um risco: crescendo do teto, elas agarram qualquer coisa que esteja abaixo — druida, árvores ou pontos de interrogação. Estes são usados para representar vantagens misteriosas, e um deles deve ser conseguido antes que o jogador possa baixar uma escada para o andar seguinte.

Terceiro ato: "A tumba do mago do mal". O clímax do drama ocorre num lúgubre e escuro cemitério. As lápides marcam as muitas sepulturas de Tetragorn, devendo ser destruídas para evitar sua reencarnação. A varinha mágica tem poder bastante para destruir cada renascimento, e as próprias lápides desaparecerão se o jogador colocar o druida sobre elas. Mas a batalha entre o bem e o mal não é tão simples, pois todas as aranhas que escaparam no segundo ato se transformam em aranhas-zumbis e vêm em defesa de seu mestre.

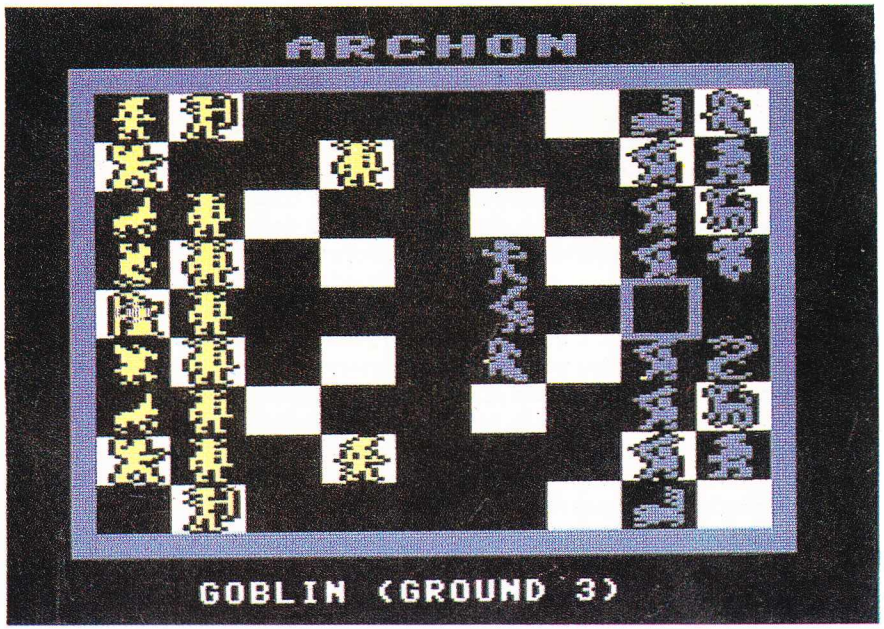
É imperioso reservar bastante energia para a tumba, pois aqui o jogador deve manter-se muito atento. O último ato pode revelar-se extremamente frustrante, por ser impossível chegar até ele sem passar pelas partes iniciais do jogo.

Das duas versões do jogo, a implementação do Atari é a melhor. A do Commodore mostra-se mais lenta, além de conter efeitos gráficos e sonoros um pouco menos espetaculares. Os dois modelos podem ser prejudicados por joysticks de respostas abruptas ou lentas. Mas, à parte esses problemas, os proprietários de qualquer das versões mencionadas certamente acharão o jogo magnífico e encantador — ainda que caro.

Necromancer: Para os computadores Atari, Commodore 64 e Apple II.
Editor: Synapse Software, Estados Unidos.
Autores: Bill Williams (Atari), Scott e Steve Coleman (Commodore 64).
Joysticks: Padrões Atari e Commodore.
Formato: Disco e cassete.

PONTOS DE ENERGIA

Archon (análise)



Combinando elementos de xadrez com gráficos de movimento rápido de fliperamas, o Archon, da Ariolasoft, exige destreza do jogador, como estrategista e manipulador de joysticks.

Esse jogo tanto pode fascinar o enxadrista como satisfazer o mais fanático entusiasta dos videojogos. A base da competição consiste na batalha entre as forças da Luz e das Trevas.

Tudo começa com a apresentação de um tabuleiro estratégico, no qual os dois grupos adversários são posicionados em linhas e colunas, tudo de forma muito semelhante ao xadrez.

As peças chamam-se "fênix" e "cavaleiro", pelo lado da Luz, e "agoureira", "gnomos" e "dragões", pelo lado das Trevas. Os ícones que representam os elementos de cada um dos lados têm força e capacidade de movimentação diferentes. Algumas peças, por exemplo, podem voar (saltam sobre as oponentes à sua frente), enquanto outras só se movimentam no chão.

O próprio tabuleiro se divide como uma matriz de 9 x 9. À primeira vista se parece muito com um tabuleiro de

Perfilando
Pode-se descrever o Archon como um jogo de xadrez animado. Ao ser movimentada para outra casa do tabuleiro, uma peça pode marchar, serpentear ou voar, dependendo de suas características. Efetuado o movimento, se a casa estiver ocupada por uma peça adversária, a tela passará para uma ampliação da casa, e a batalha terá início.